Изобретая синхронную репликацию

Владислав Шпилевой



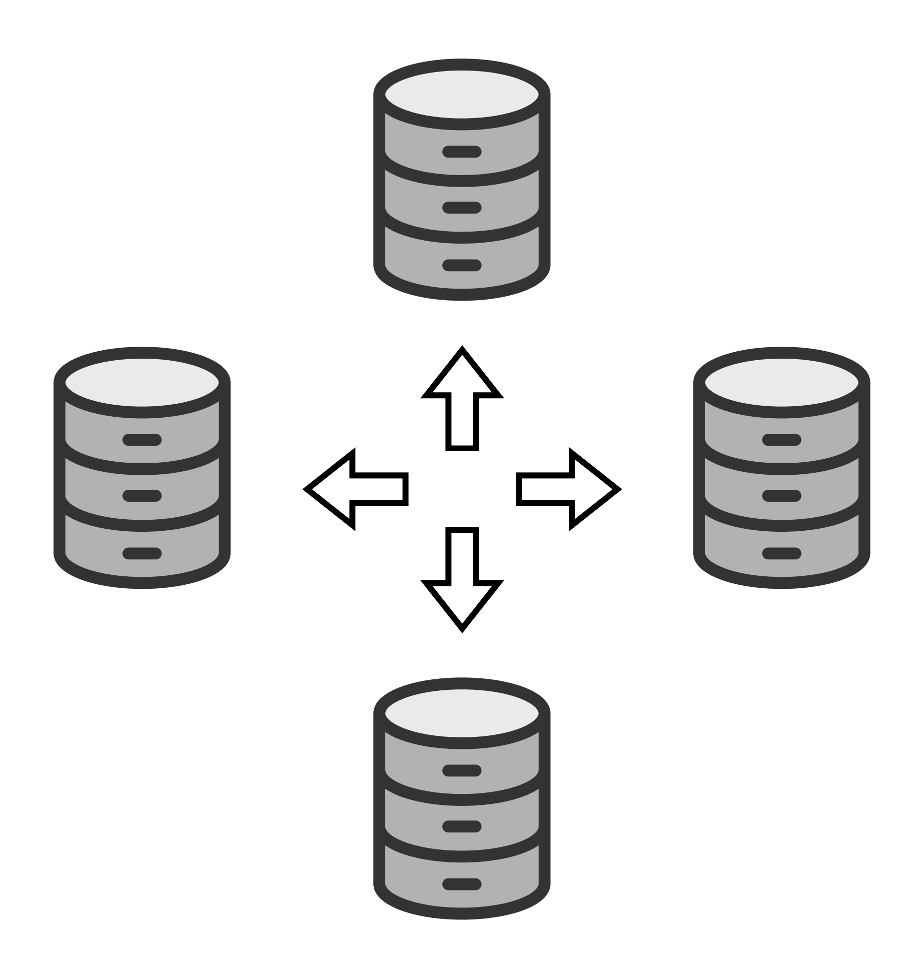


План доклада

- ✔ План доклада
- Репликация
- История репликации в Tarantool
- Алгоритм Raft
- Транзакции в Tarantool
- Синхронная репликация
- Интерфейс
- Отличия от Raft
- Планы



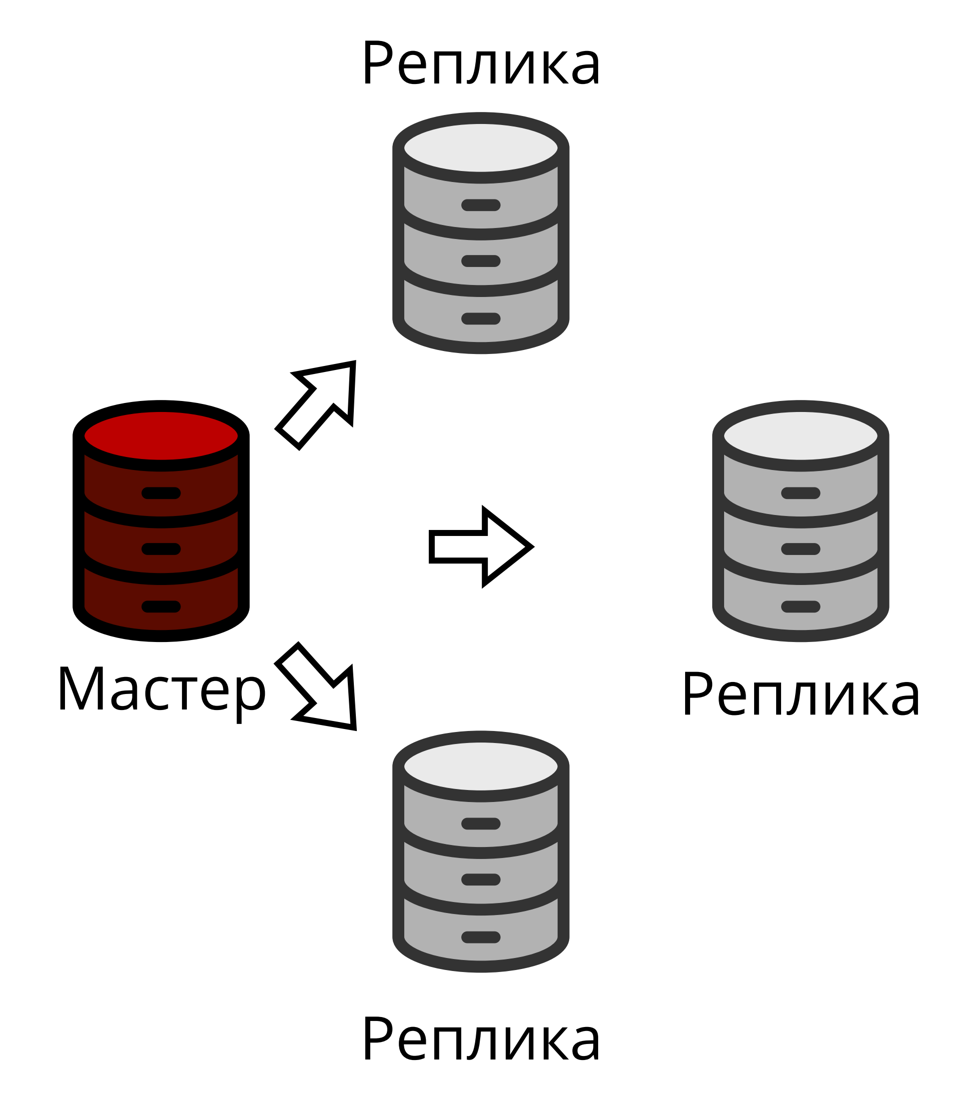
Репликация [1]



Репликационная группа –
"репликасет"



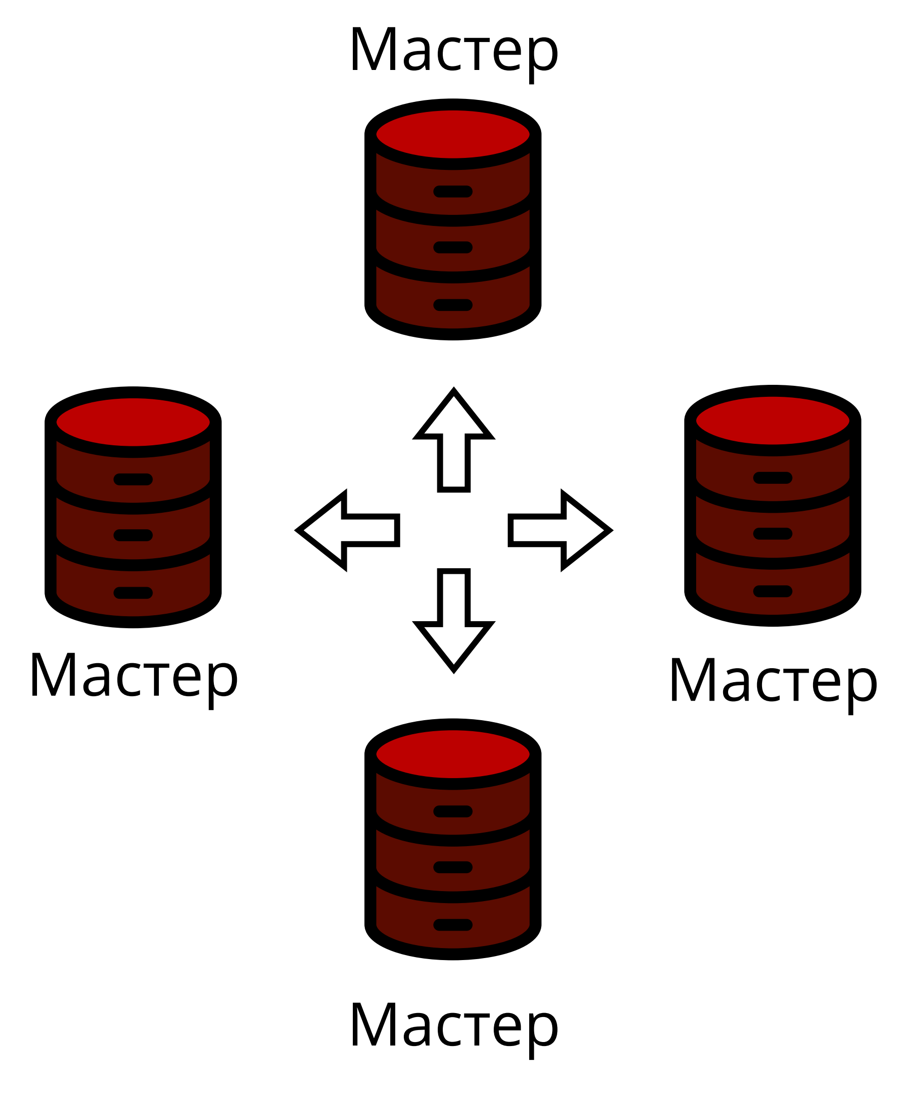
Репликация [1]



Репликационная группа –
"репликасет"



Репликация [1]



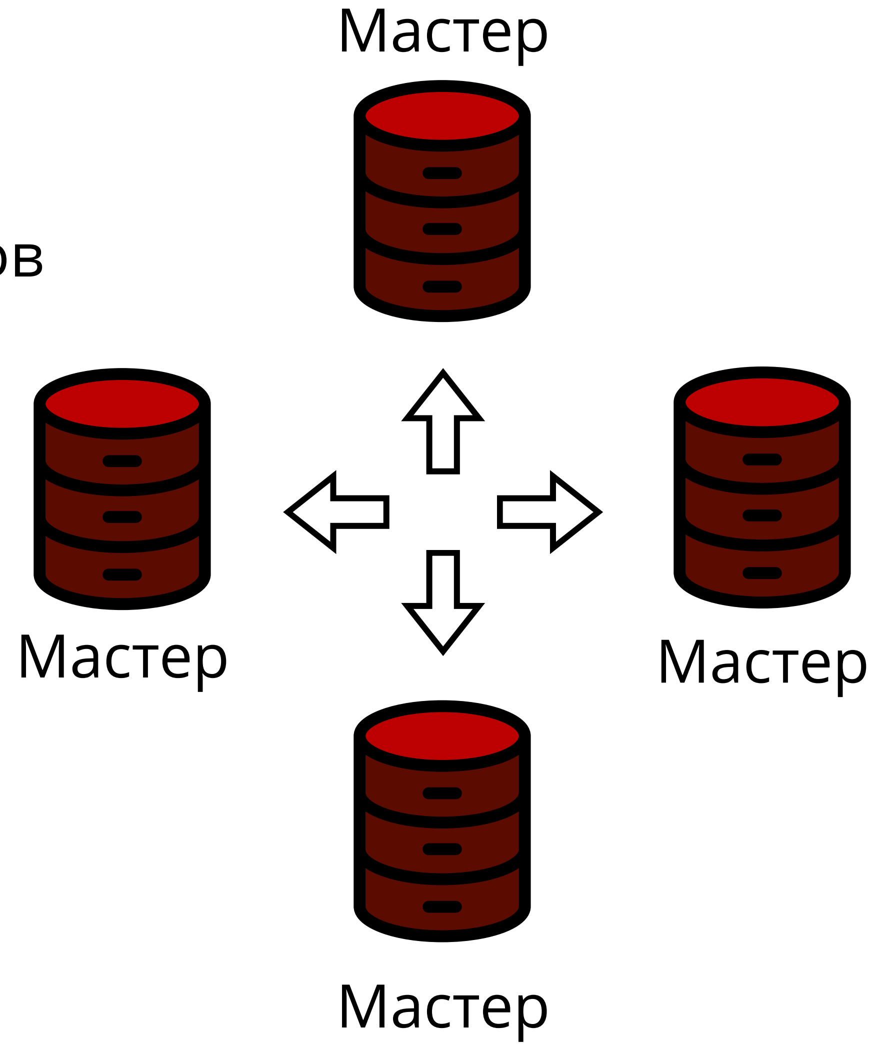
Репликационная группа – "репликасет"



Репликация [1]

Задачи

- Резервная копия
- Балансировка запросов



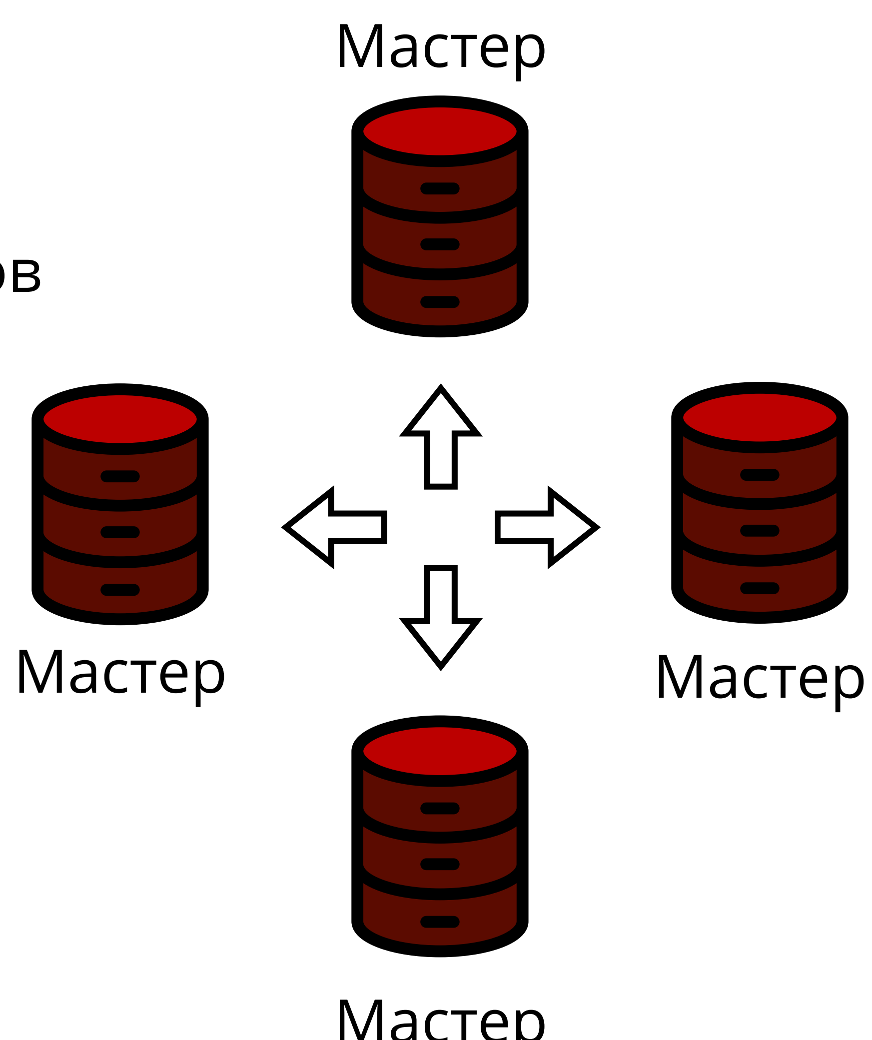
Репликационная группа –
"репликасет"



Репликация [1]

Задачи

- Резервная копия
- Балансировка запросов



Типы

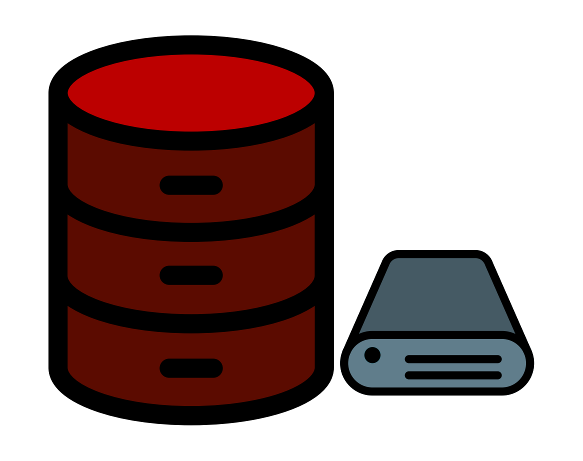
- Асинхронная
- Синхронная

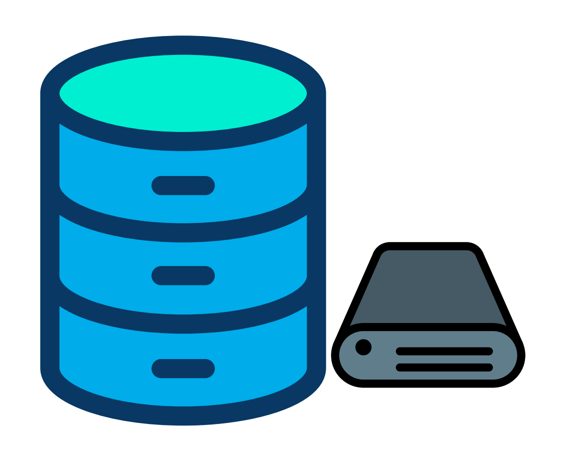
Репликационная группа – "репликасет"

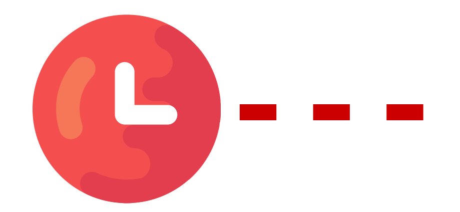


Репликация [2]: асинхронная

Запись в журнал – успешный коммит. **Не ждет репликацию**



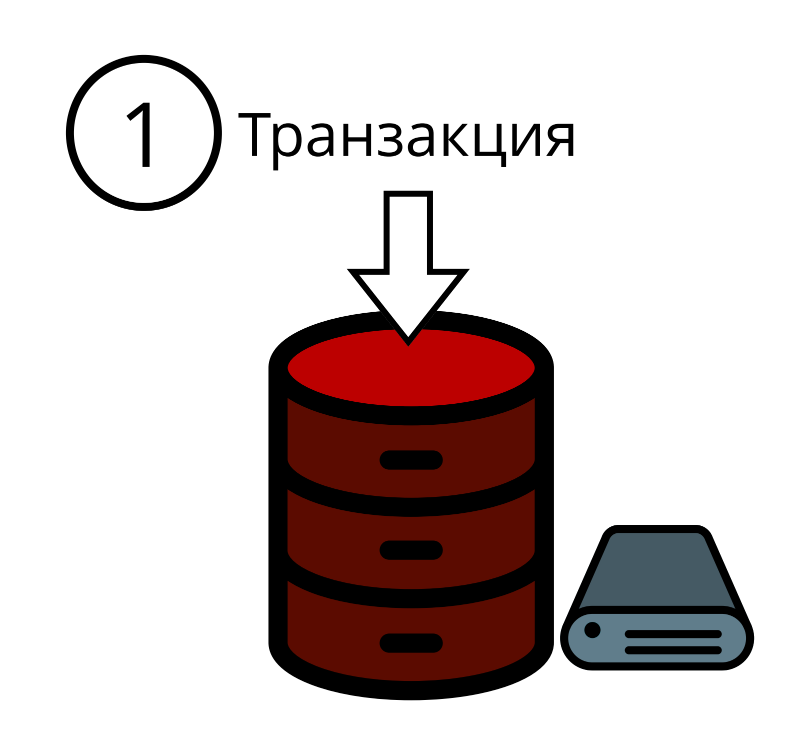


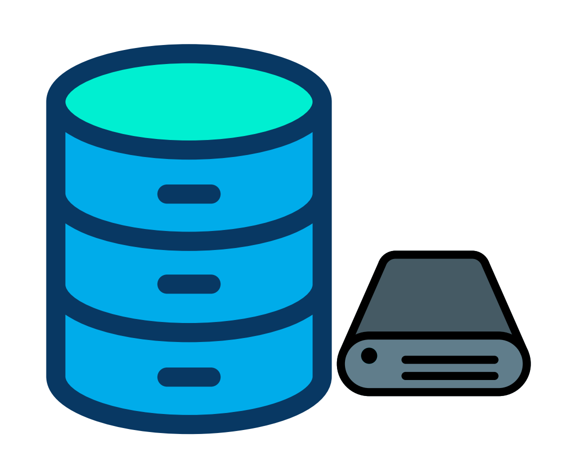




Репликация [2]: асинхронная

Запись в журнал – успешный коммит. **Не ждет репликацию**



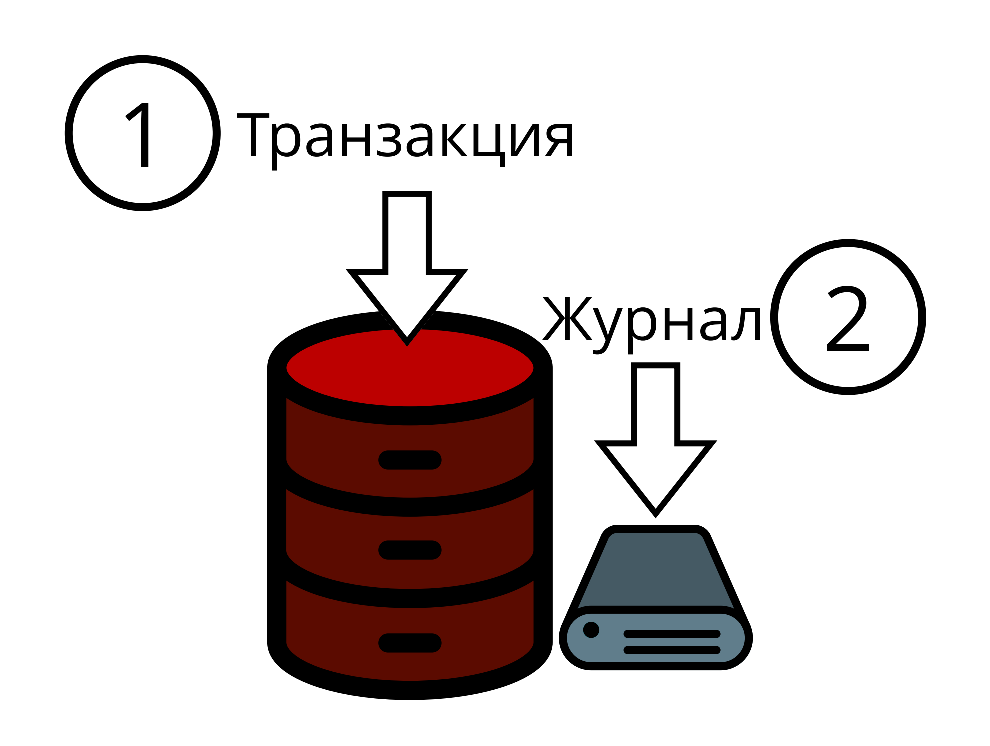


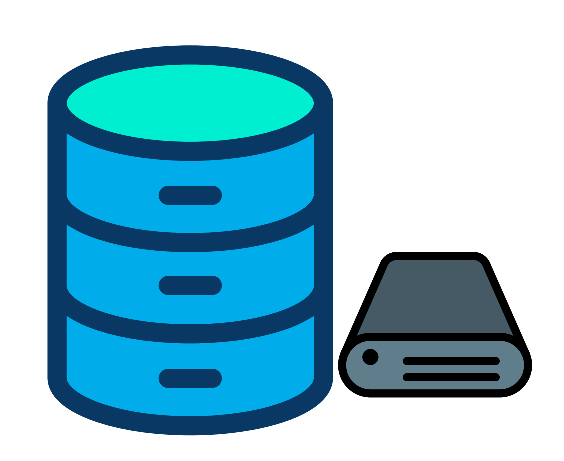




Репликация [2]: асинхронная

Запись в журнал – успешный коммит. **Не ждет репликацию**



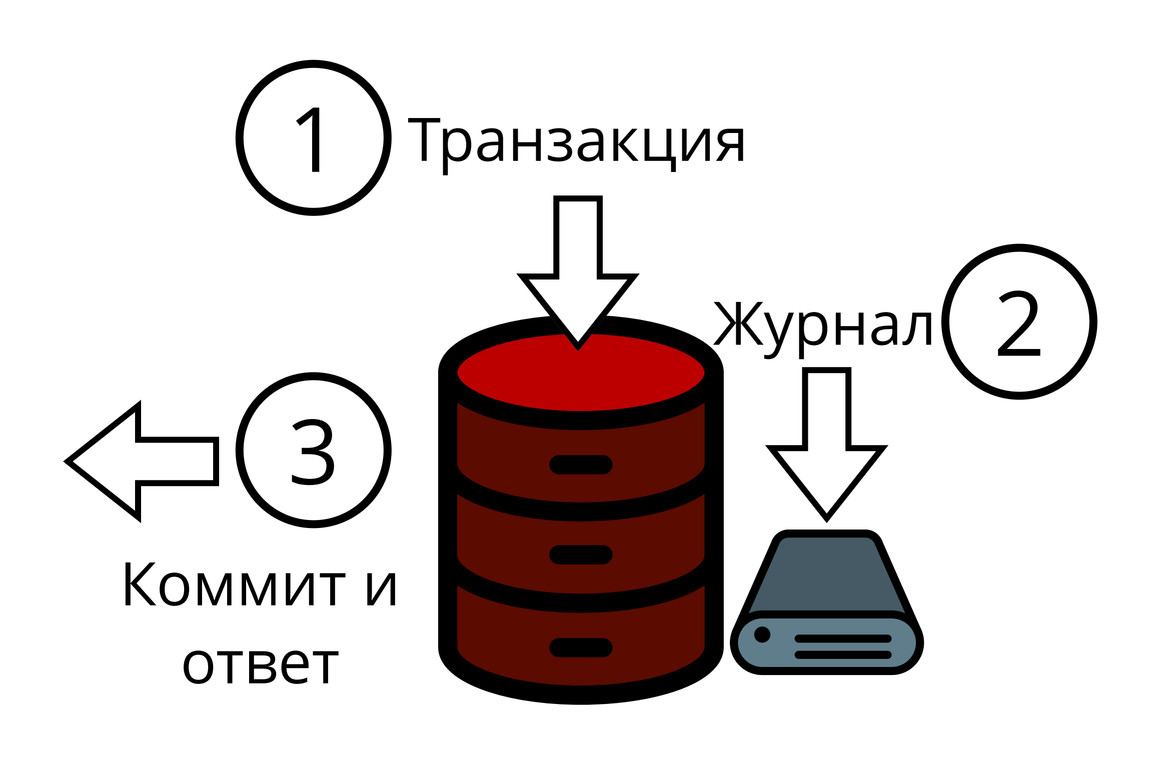


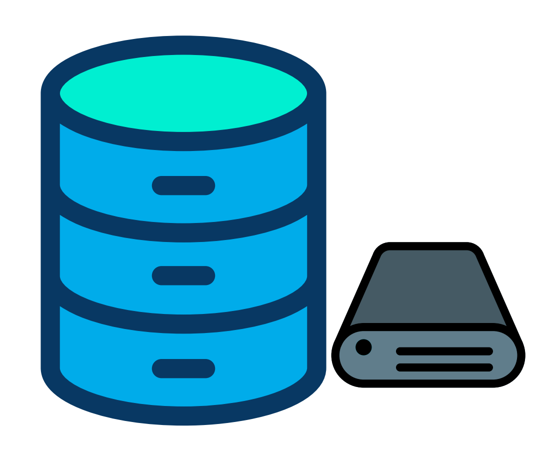




Репликация [2]: асинхронная

Запись в журнал – успешный коммит. **Не ждет репликацию**







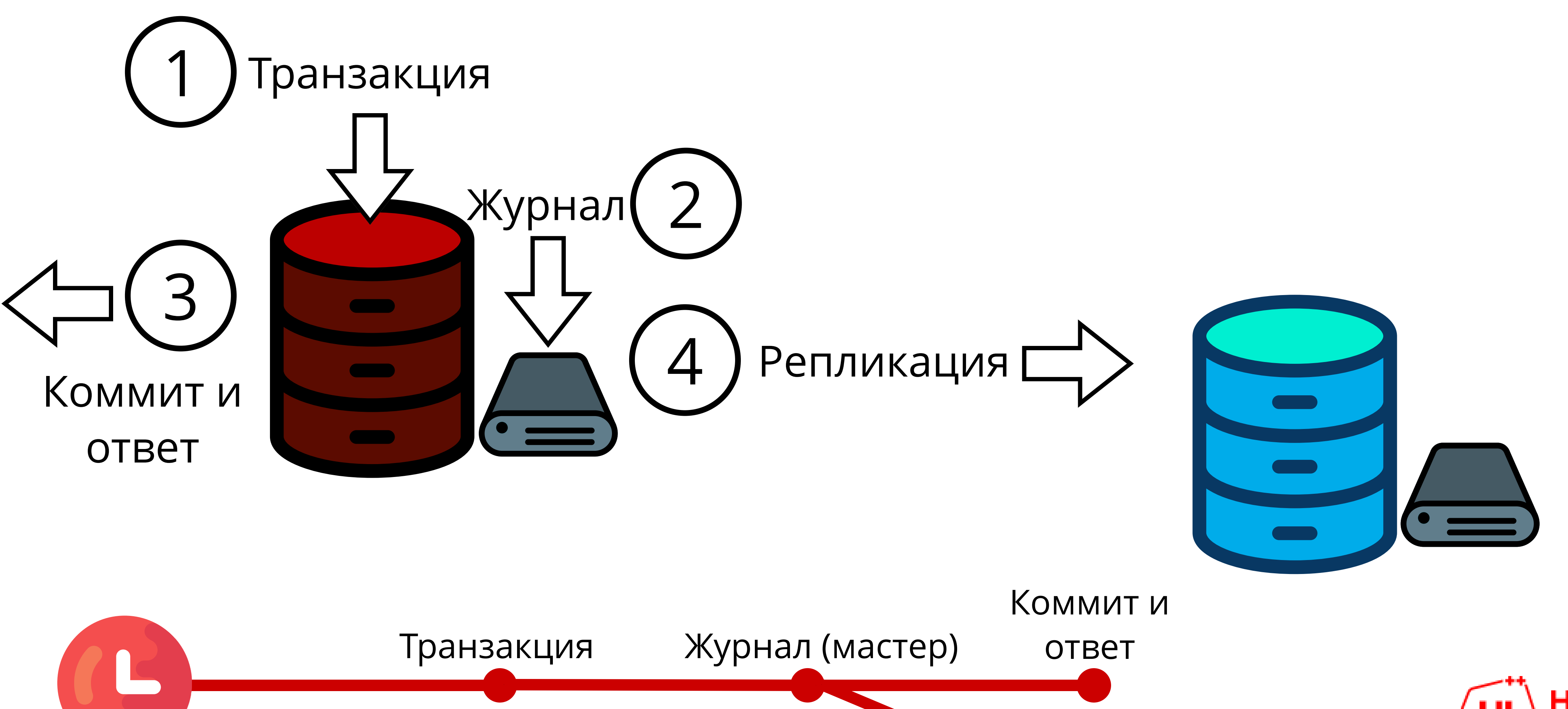
Транзакция

Журнал (мастер)

Коммит и ответ

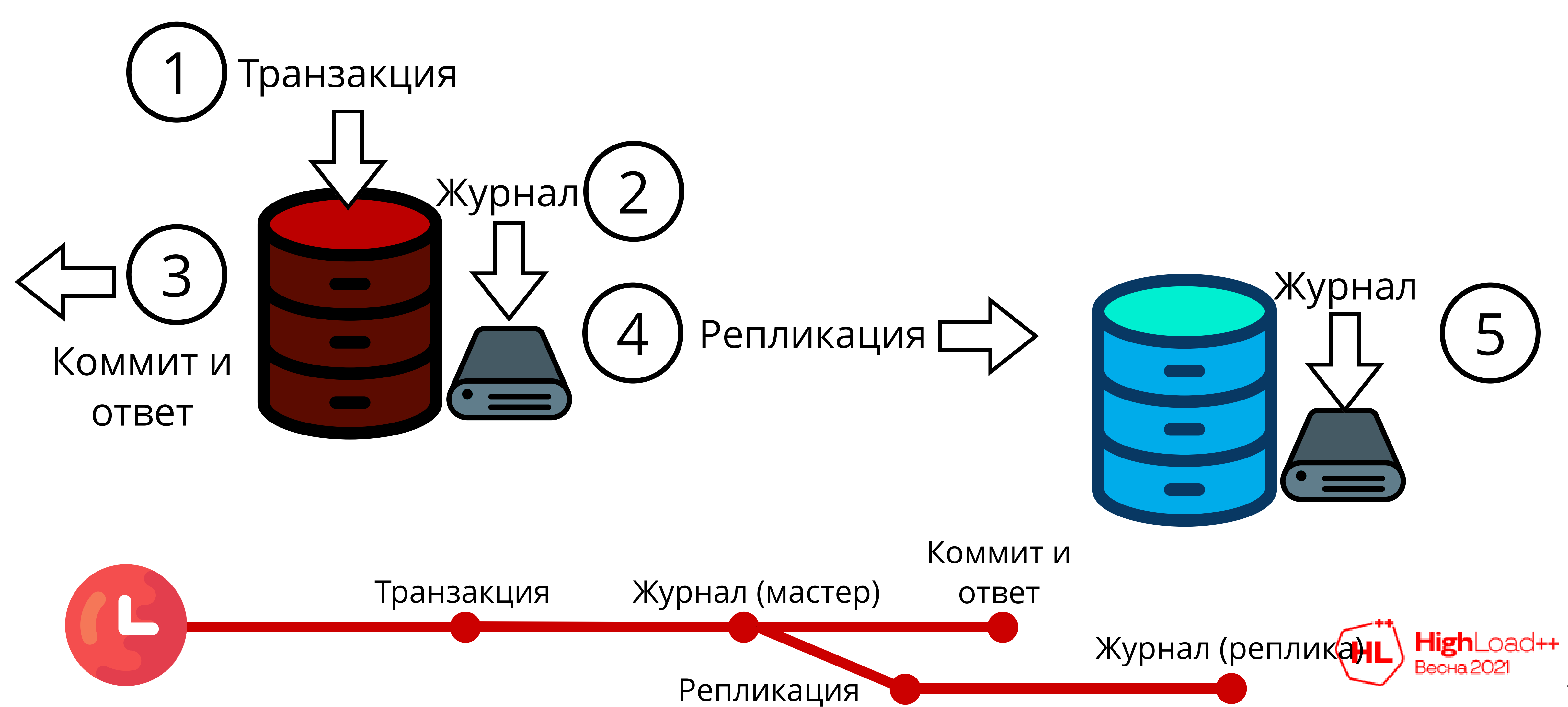
Репликация [2]: асинхронная

Запись в журнал – успешный коммит. **Не ждет репликацию**



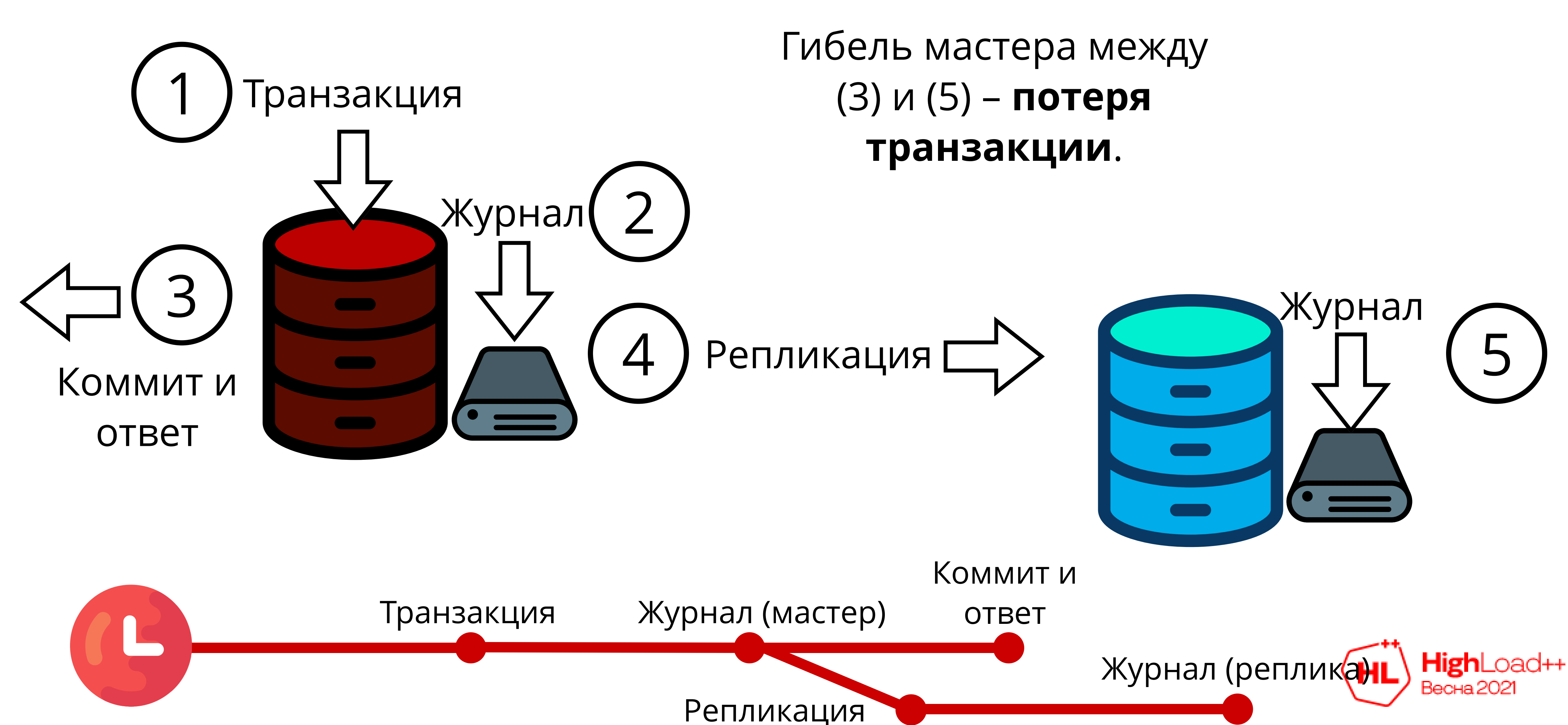
Репликация [2]: асинхронная

Запись в журнал – успешный коммит. **Не ждет репликацию**



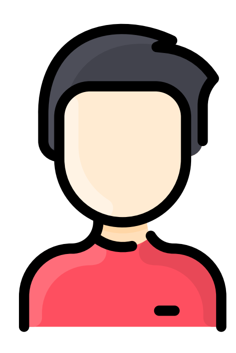
Репликация [2]: асинхронная

Запись в журнал – успешный коммит. **Не ждет репликацию**

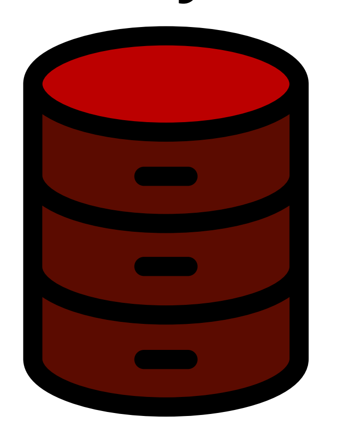


Репликация [3]: потеря данных

У меня 100 денег

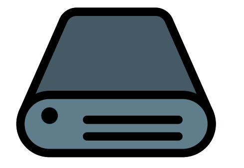


{money: 100}



{money: 100}

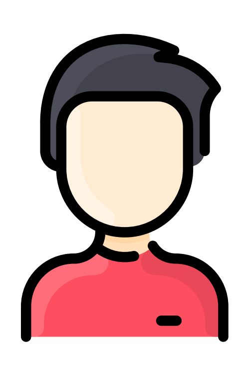




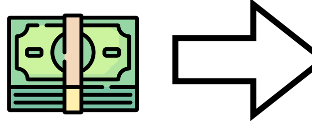


Репликация [3]: потеря данных

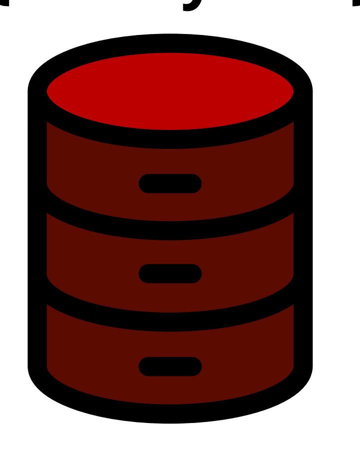
Кладу еще 50



{money: +50}

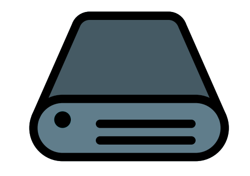


{money: 100}



{money: 100}

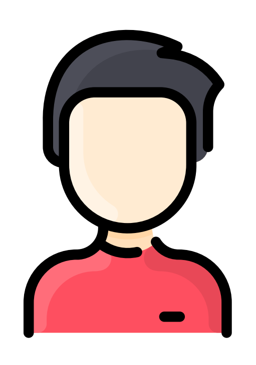


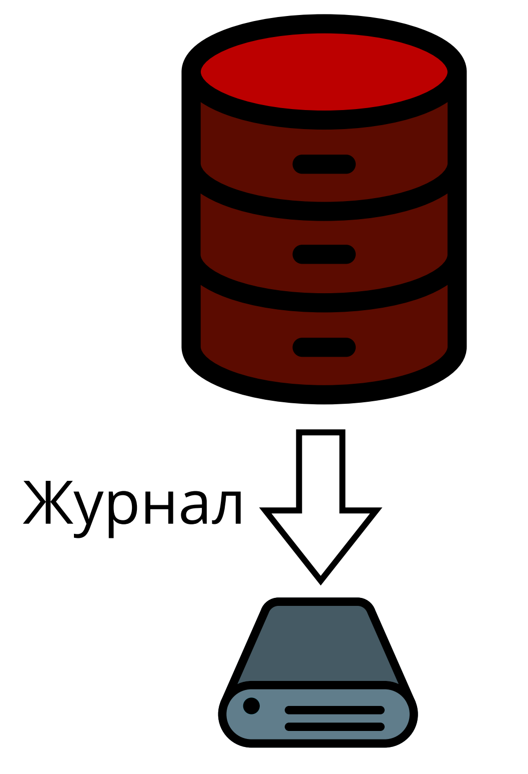




Репликация [3]: потеря данных

Кладу еще 50









Репликация [3]: потеря данных

У меня 150 денег

{success}

{money: 150}

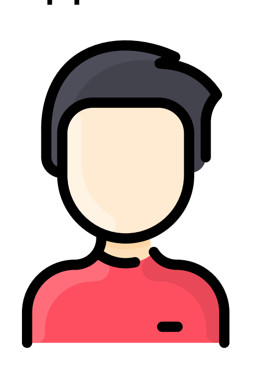
Начало репликации

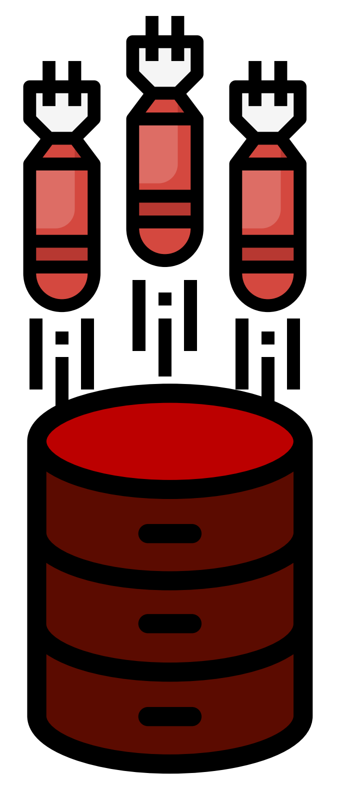
{money: 100}

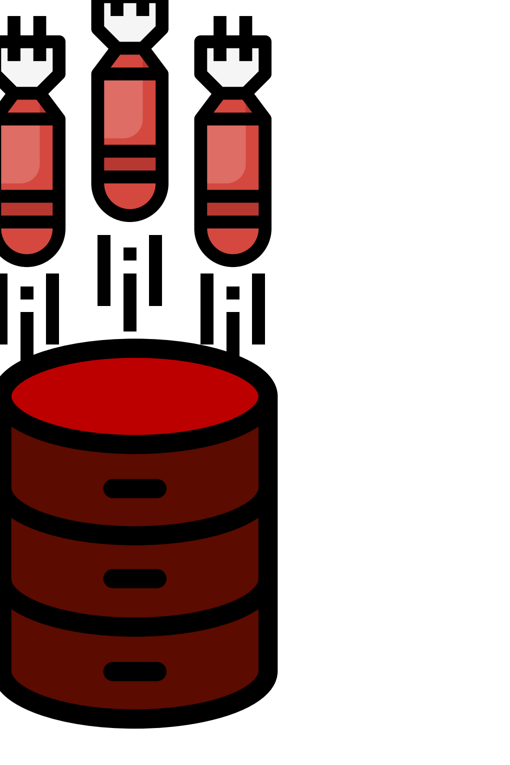


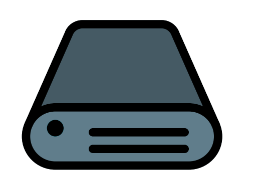
Репликация [3]: потеря данных

У меня 150 денег









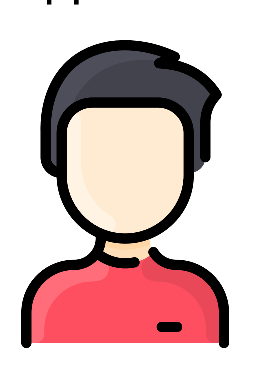






Репликация [3]: потеря данных

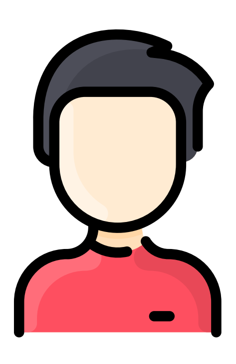
У меня 150 денег



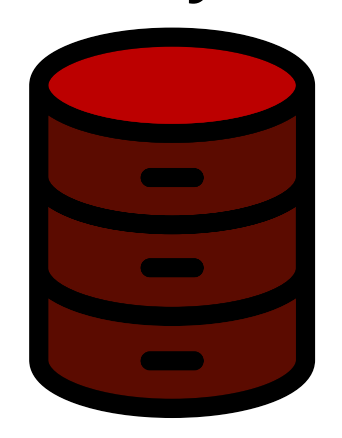
{money: 100}

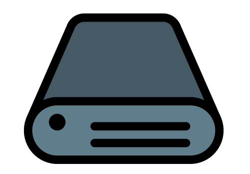
Репликация [3]: потеря данных

У меня 150 денег



{money: 100}





Репликация [3]: потеря данных

У меня 150 денег

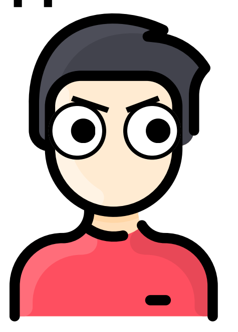
{money: 100}

get({money})



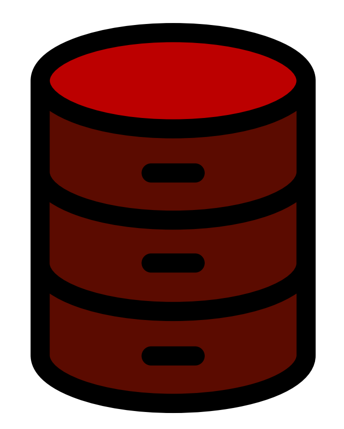
Репликация [3]: потеря данных

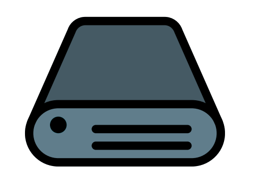
Где мои 150 денег?!



{money: 100}

{money: 100}





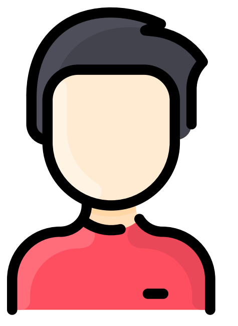
При асинхронной репликации гарантии – почти **как без репликации**

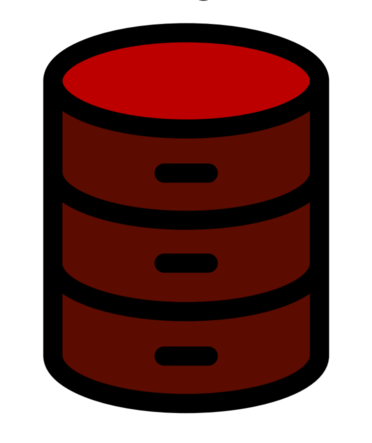
Репликация [4]: грязные чтения

Ждать репликацию **вручную** – видеть **незакоммиченные** данные

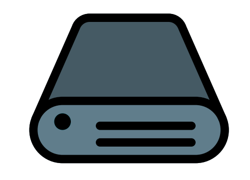
{money: 100}

У меня 100 денег







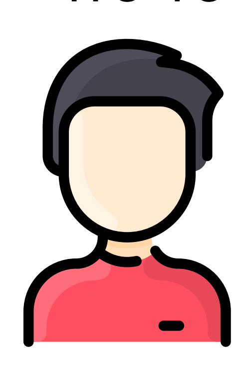


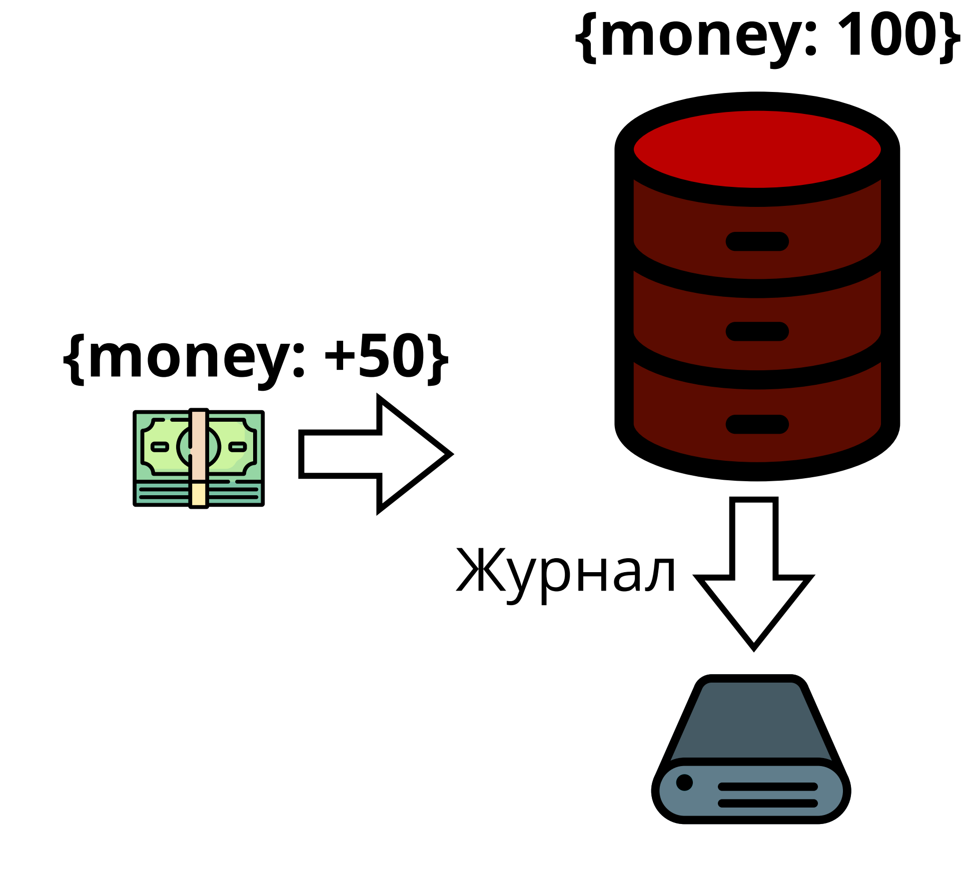


Репликация [4]: грязные чтения

Ждать репликацию **вручную** – видеть **незакоммиченные** данные

Кладу еще 50, чтобы купить что-то



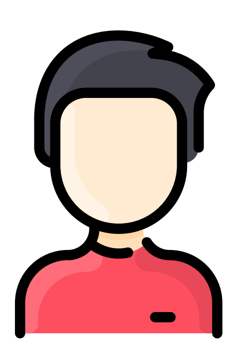




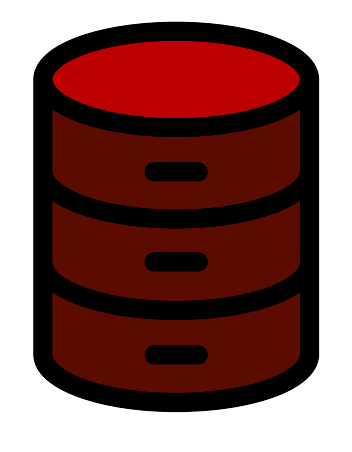
Репликация [4]: грязные чтения

Ждать репликацию **вручную** – видеть **незакоммиченные** данные

Жду репликации, но **изменения уже видимы**

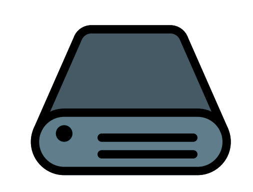












Репликация [4]: грязные чтения

Ждать репликацию **вручную** – видеть **незакоммиченные** данные

{money: -150}

Покупаю на "грязные" деньги

{money: 150}

Ожидание репликации

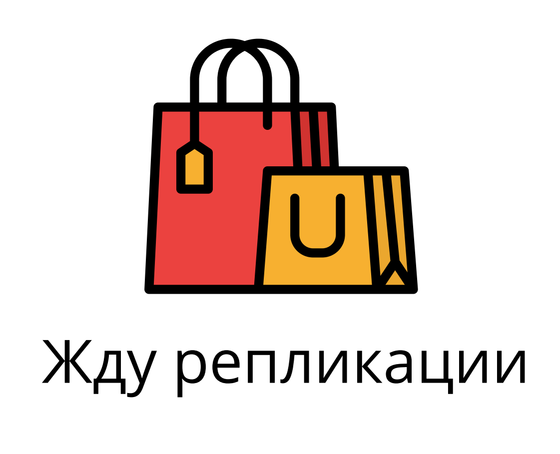
Журнал

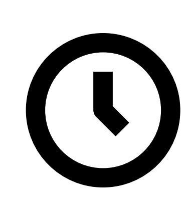
Репликация сломалась



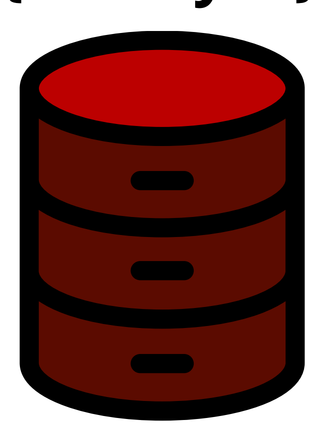
Репликация [4]: грязные чтения

Ждать репликацию **вручную** – видеть **незакоммиченные** данные



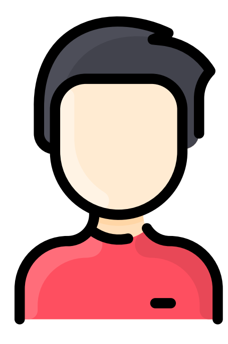




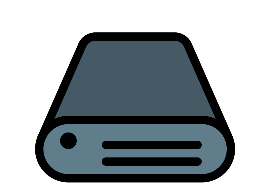


Репликация сломалась





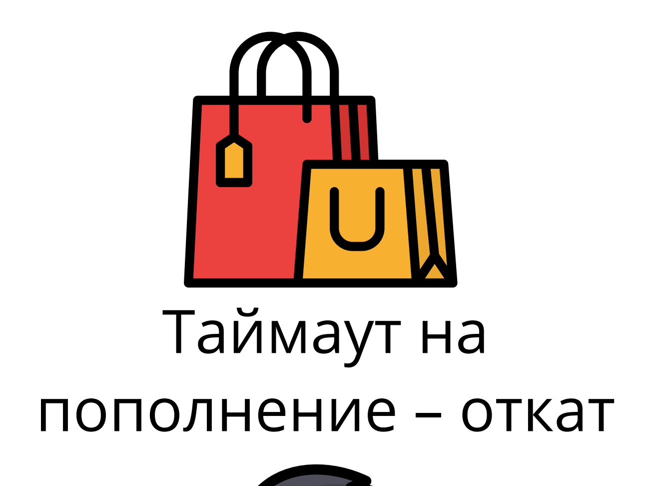


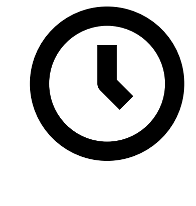




Репликация [4]: грязные чтения

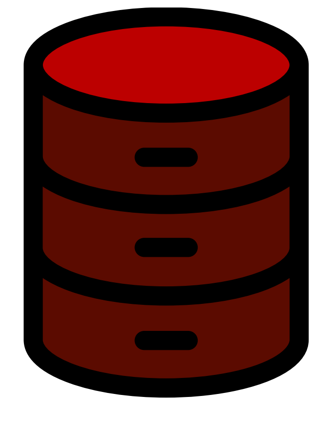
Ждать репликацию **вручную** – видеть **незакоммиченные** данные





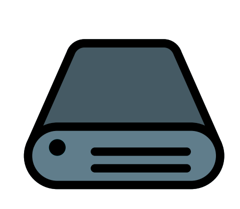














Репликация [4]: грязные чтения

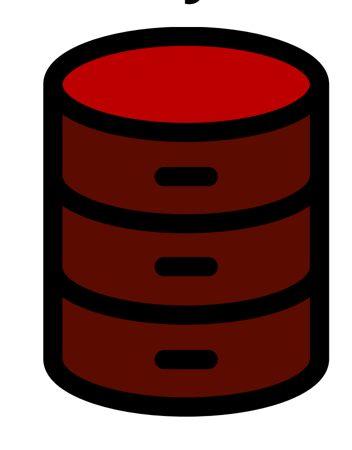
Ждать репликацию **вручную** – видеть **незакоммиченные** данные



А покупка реплицировалась

{success}

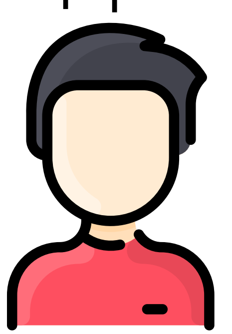
{money: 100}



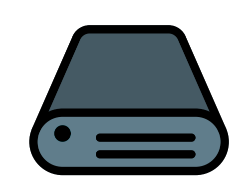


Репликация работает





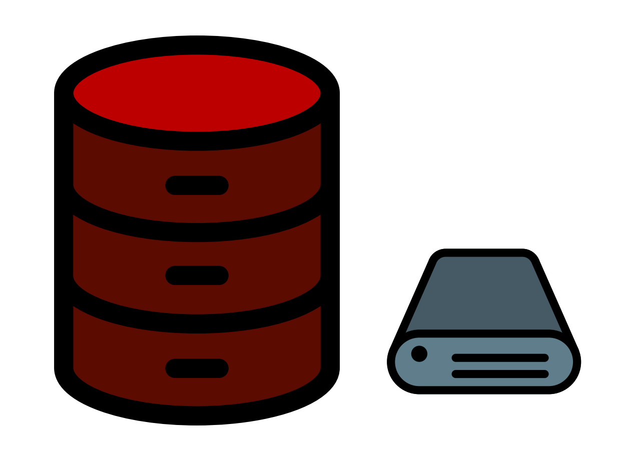
Из-за грязного чтения покупка прошла **бесплатно** (это плохо)!

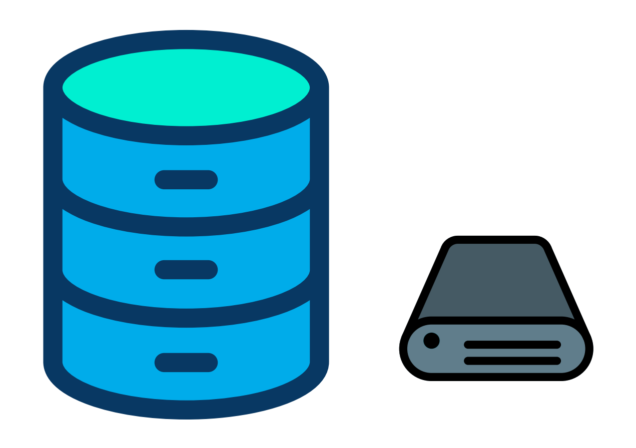


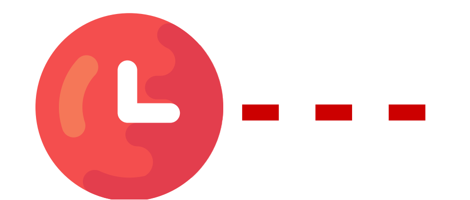


Синхронная репликация [1]

Репликация **гарантируется до коммита**





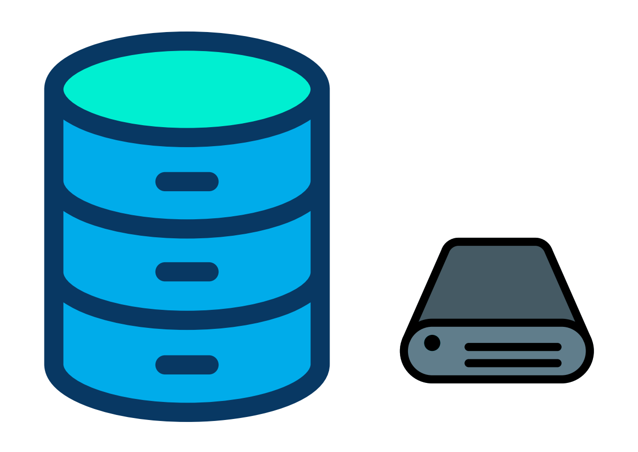




Синхронная репликация [1]

Репликация **гарантируется до коммита**



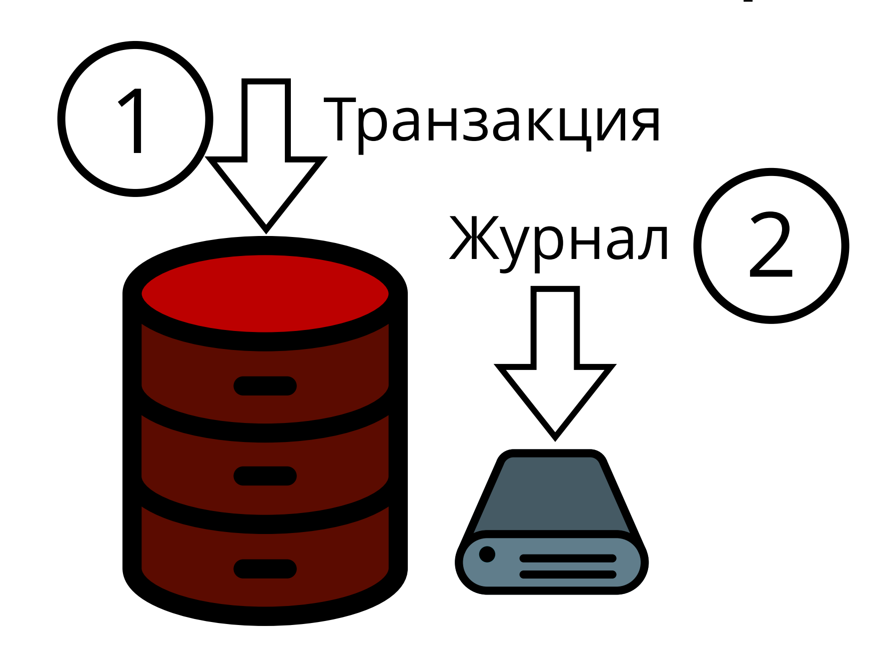


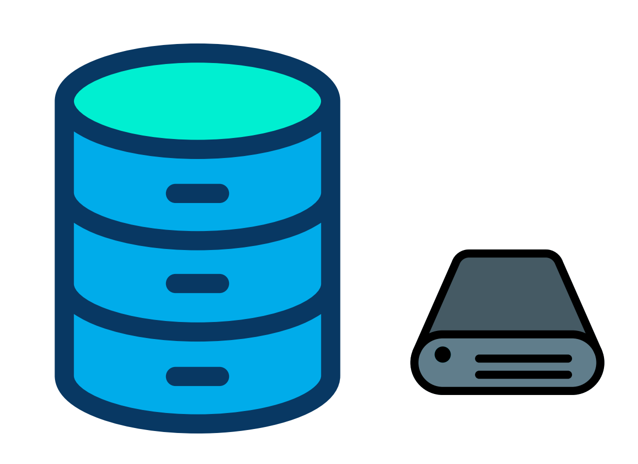




Синхронная репликация [1]

Репликация **гарантируется до коммита**









Синхронная репликация [1]

Репликация **гарантируется до коммита**

1 Транзакция

Журнал 2

3 Репликация





Синхронная репликация [1]

Репликация **гарантируется до коммита**

1 Транзакция

Журнал 2

3 Репликация

Журнал 4





Синхронная репликация [1]

Репликация **гарантируется до коммита**

1 Транзакция

Журнал 2

3 Репликация

Журнал 4

5 Коммит и ответ



Транзакция

Журнал (мастер)

Репликация

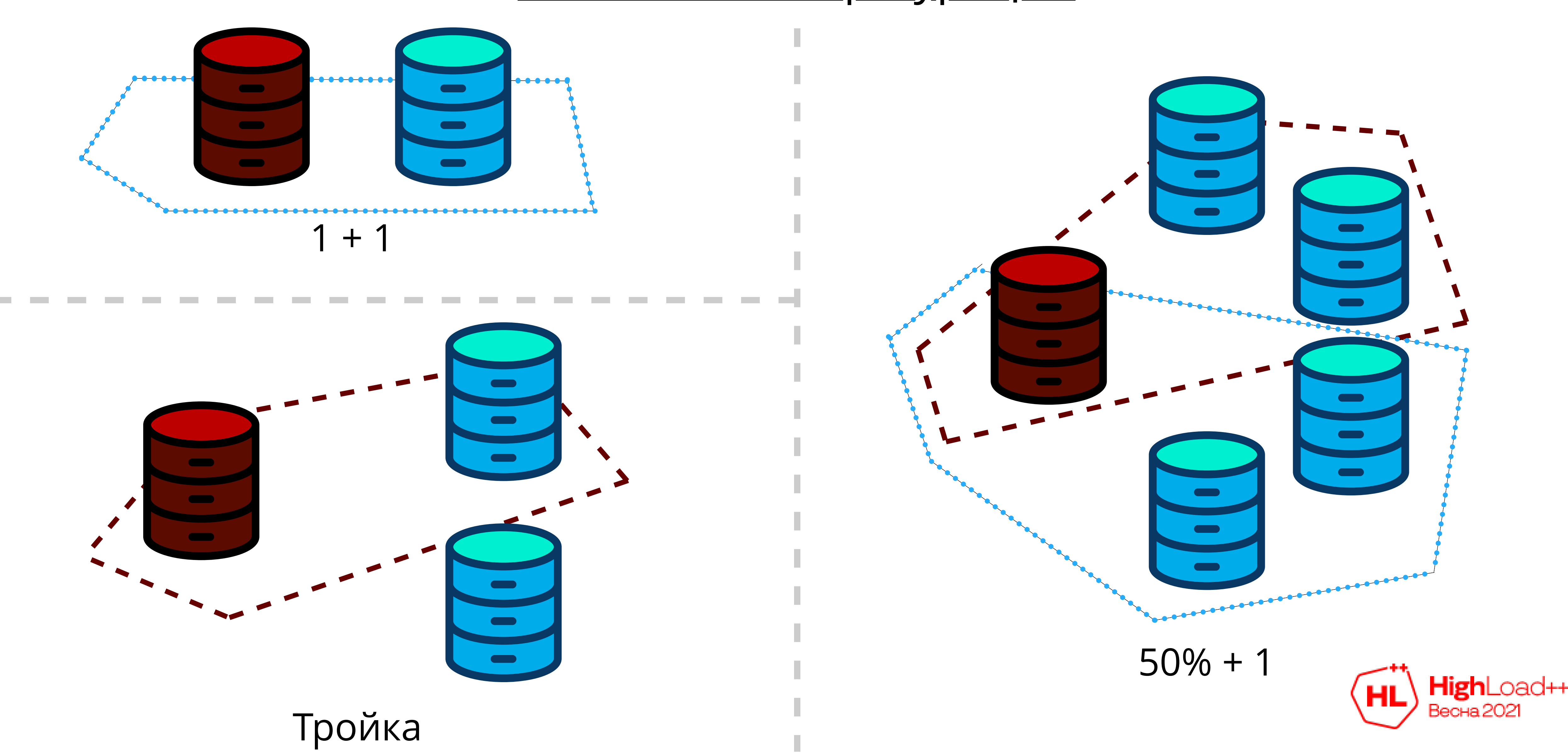
Журнал (реплика)

Коммит и ответ



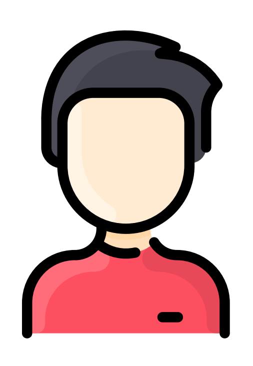
Синхронная репликация [2]: кворум

Типичные конфигурации

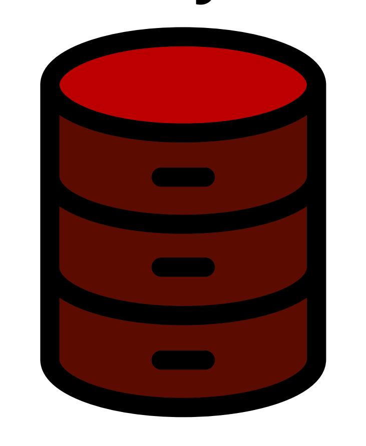


Синхронная репликация [3]

У меня 100 денег

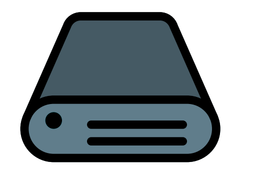


{money: 100}



{money: 100}

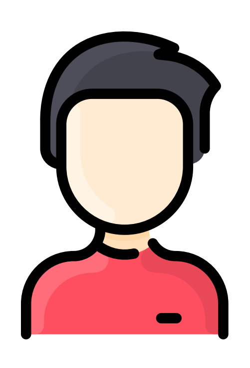




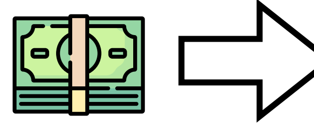


Синхронная репликация [3]

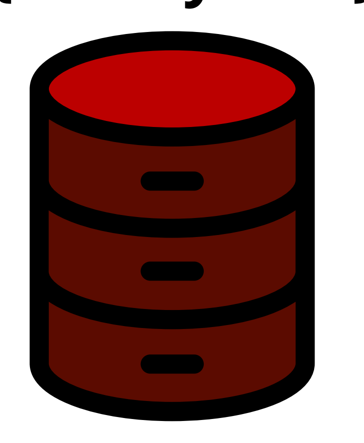
Кладу еще 50



{money: +50}

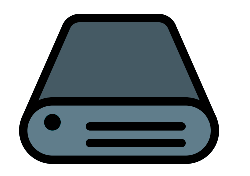


{money: 100}



{money: 100}

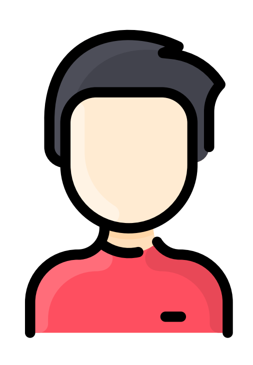




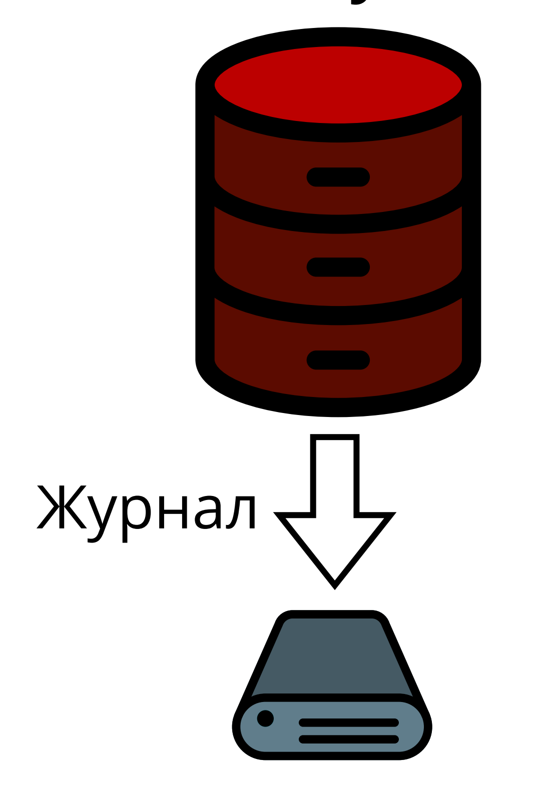


Синхронная репликация [3]

Кладу еще 50

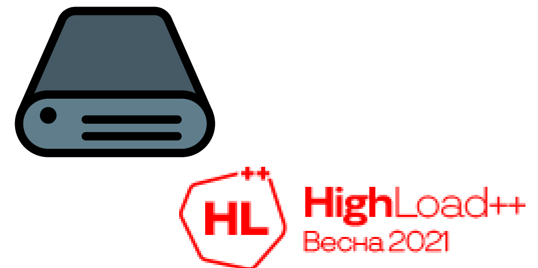


{money: 100}



{money: 100}

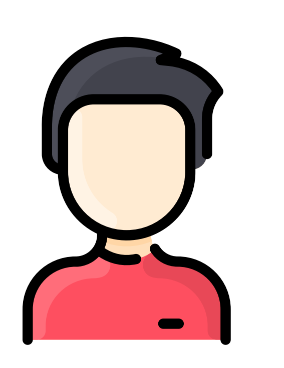




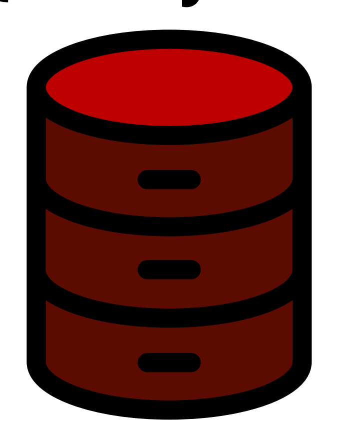
Синхронная репликация [3]

Репликация происходит
до коммита

Кладу еще 50



{money: 100}

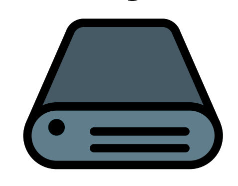


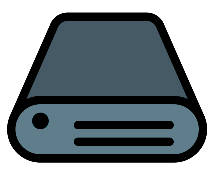
Репликация

{money: 100}



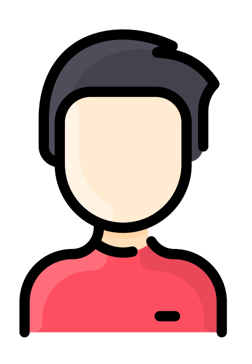
{money: 150}



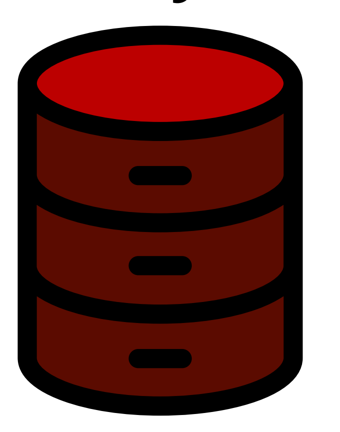


Синхронная репликация [3]

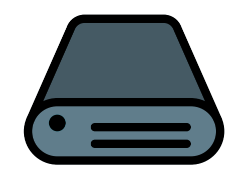
Кладу еще 50



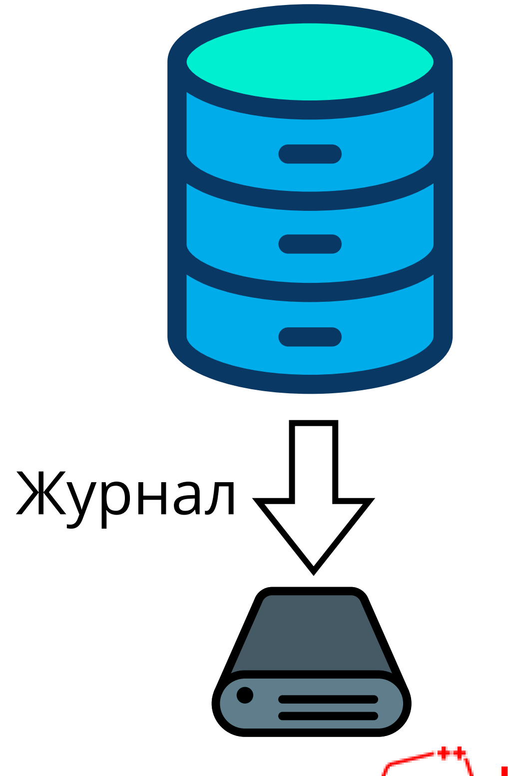
{money: 100}



{money: 150}

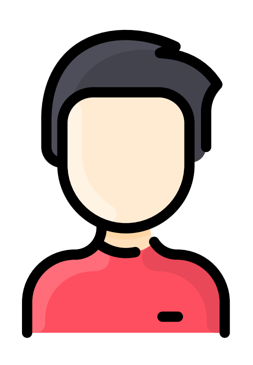


{money: 100}

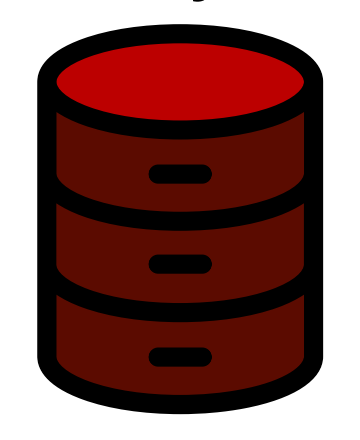


Синхронная репликация [3]

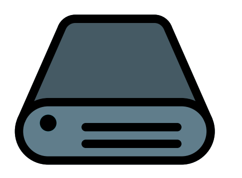
Кладу еще 50



{money: 100}



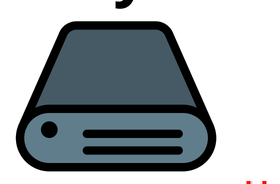
{money: 150}



{money: 100}



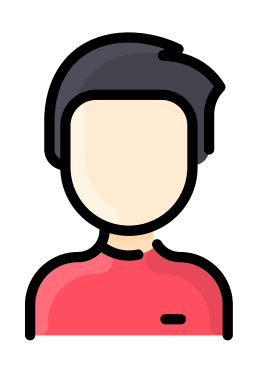
{money: 150}



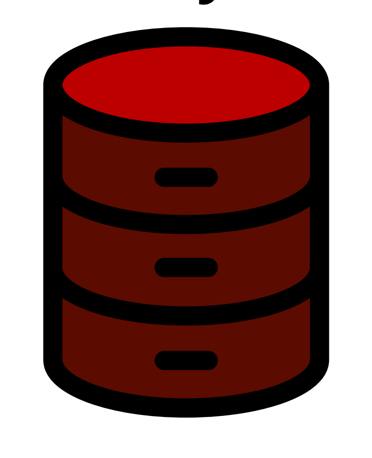


Синхронная репликация [3]

Кладу еще 50



{money: 100}

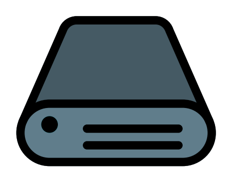




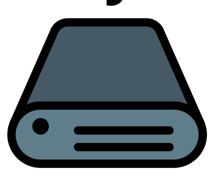
{money: 100}

Ответ

{money: 150}



{money: 150}

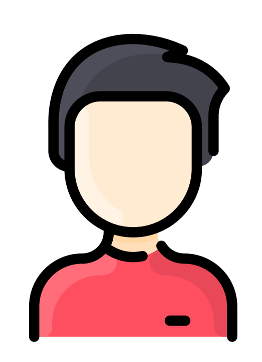


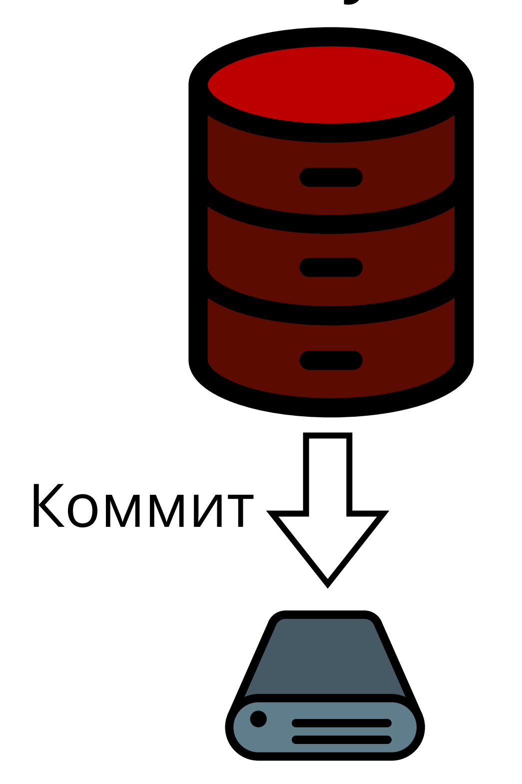


Синхронная репликация [3]

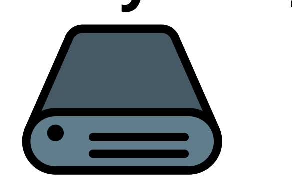
Коммит происходит
после подтверждений
от кворума реплик

Кладу еще 50







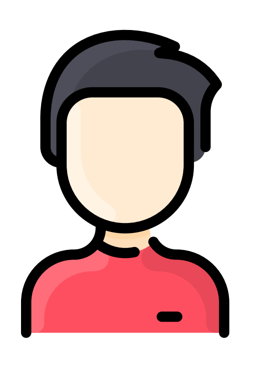




Синхронная репликация [3]

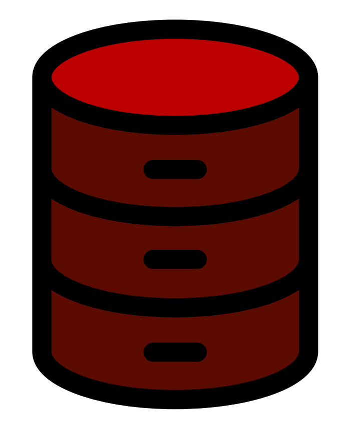
Сам факт коммита реплицируется асинхронно, но его потеря уже не критична

Кладу еще 50





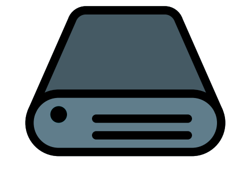




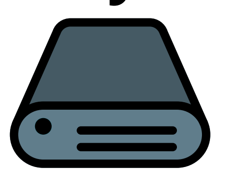


{money: 100}





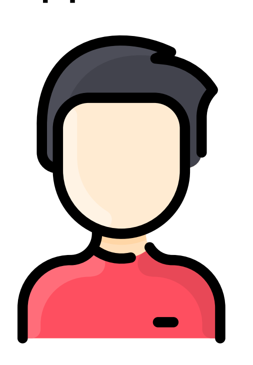


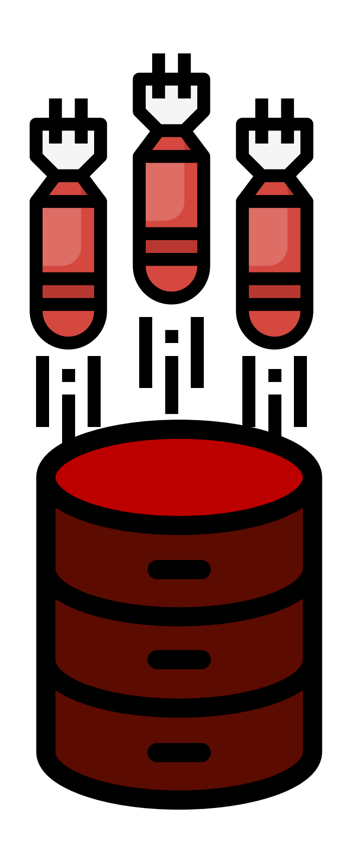


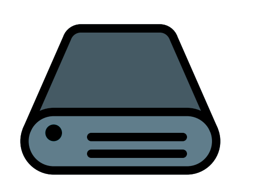


Синхронная репликация [3]

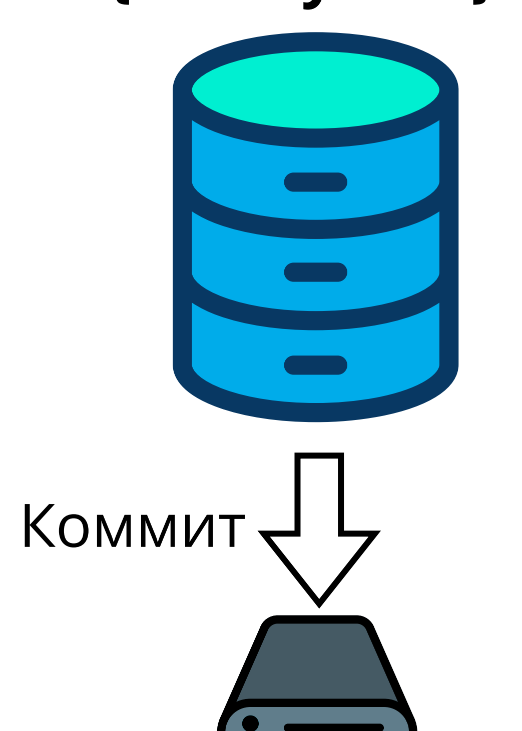
У меня 150 денег







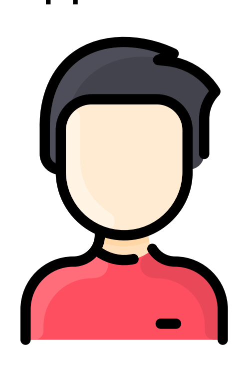
{money: 100}





Синхронная репликация [3]

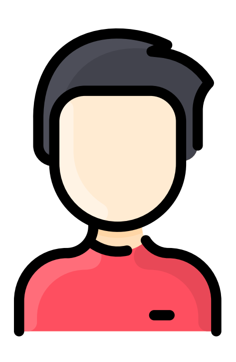
У меня 150 денег



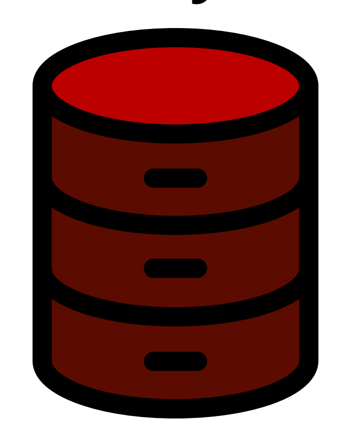
{money: 150}

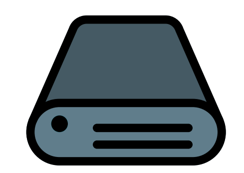
Синхронная репликация [3]

У меня 150 денег



{money: 150}





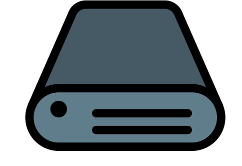
HL HighLoad++ Весна 2021

Синхронная репликация [3]

У меня 150 денег

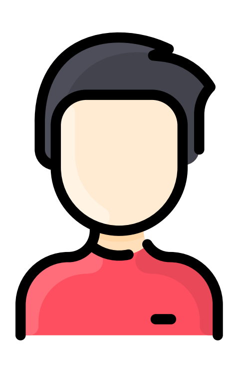
get({money})

{money: **150**}



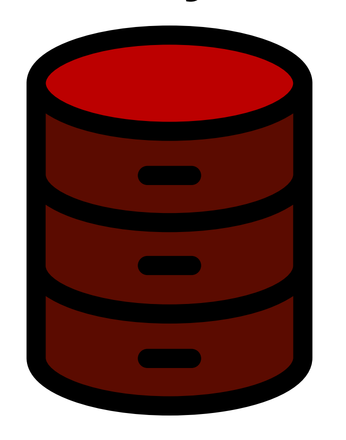
Синхронная репликация [3]

У меня 150 денег

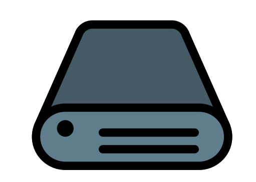


{money: 150}

{money: 150}



Синхронная репликация **гарантирует** сохранность данных, пока достаточное количество узлов живо





Сравнение

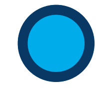
Асинхронная репликация

Синхронная репликация

HighLoad++ Весна 2021

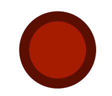
Сравнение

Асинхронная репликация



Быстрая

Синхронная репликация



Медленная

Сравнение

Асинхронная репликация

- **Быстрая**
- **Высокая** доступность

Синхронная репликация

- **Медленная**
- **Хрупкая** доступность

Сравнение

Асинхронная репликация

- **Быстрая**
- **Высокая** доступность
- **Легко** конфигурировать

Синхронная репликация

- **Медленная**
- **Хрупкая** доступность
- **Трудно** конфигурировать

Сравнение

Асинхронная репликация

- **Быстрая**
- **Высокая** доступность
- **Легко** конфигурировать
- **Есть** мастер-мастер

Синхронная репликация

- **Медленная**
- **Хрупкая** доступность
- **Трудно** конфигурировать
- **Только** мастер-реплика



Сравнение

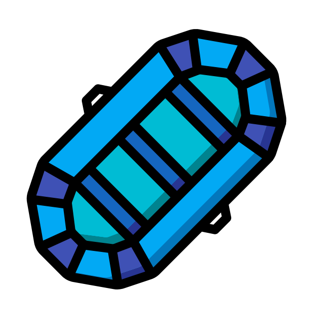
Асинхронная репликация

- **Быстрая**
- **Высокая** доступность
- **Легко** конфигурировать
- **Есть** мастер-мастер
- Легко **потерять** данные

Синхронная репликация

- **Медленная**
- **Хрупкая** доступность
- **Трудно** конфигурировать
- **Только** мастер-реплика
- **Повышенная** надежность

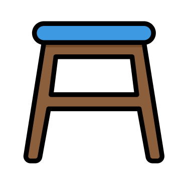
Raft



Алгоритм синхронной репликации



Гарантия сохранности данных на > 50% узлов;



Чрезвычайная простота



Проверен временем



Выборы лидера



История разработки [1]

Год 2015

История разработки [1]

Год 2015

Появился спрос на синхронность

Выбрали Raft, составили **великий** план

История разработки [1]

Год 2015

Появился спрос на синхронность

Выбрали Raft, составили **великий** план

\- -

Модуль SWIM – для сборки кластера

История разработки [1]

Год 2015

Появился спрос на синхронность

Выбрали Raft, составили **великий** план

- -

Модуль SWIM – для сборки кластера

- -

Ручные выборы – box.ctl.promote()

История разработки [1]

Год 2015

Появился спрос на синхронность

Выбрали Raft, составили **великий** план

- -

Модуль SWIM – для сборки кластера

- -

Ручные выборы – box.ctl.promote()

- -

Оптимизации репликаци

История разработки [1]

Год 2015

Появился спрос на синхронность

Выбрали Raft, составили **великий** план

- -

Модуль SWIM – для сборки кластера

- -

Ручные выборы – box.ctl.promote()

- -

Оптимизации репликаци

- -

Авто-прокси

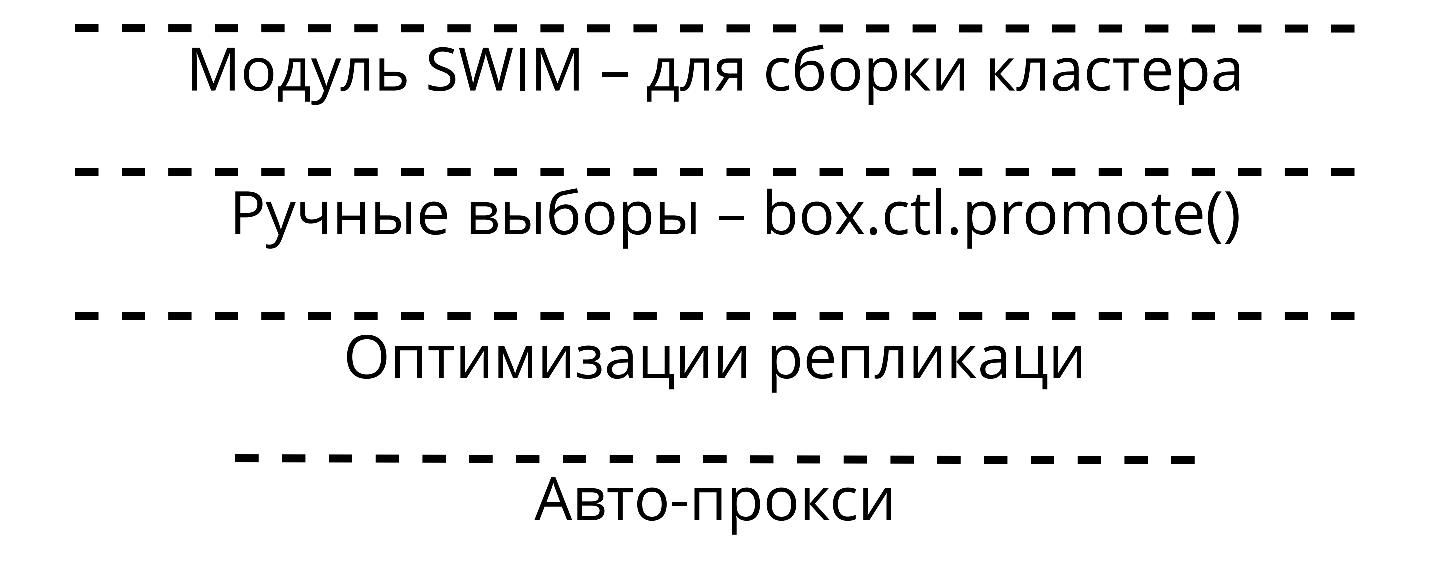


История разработки [1]

Год 2015

Появился спрос на синхронность

Выбрали Raft, составили **великий** план





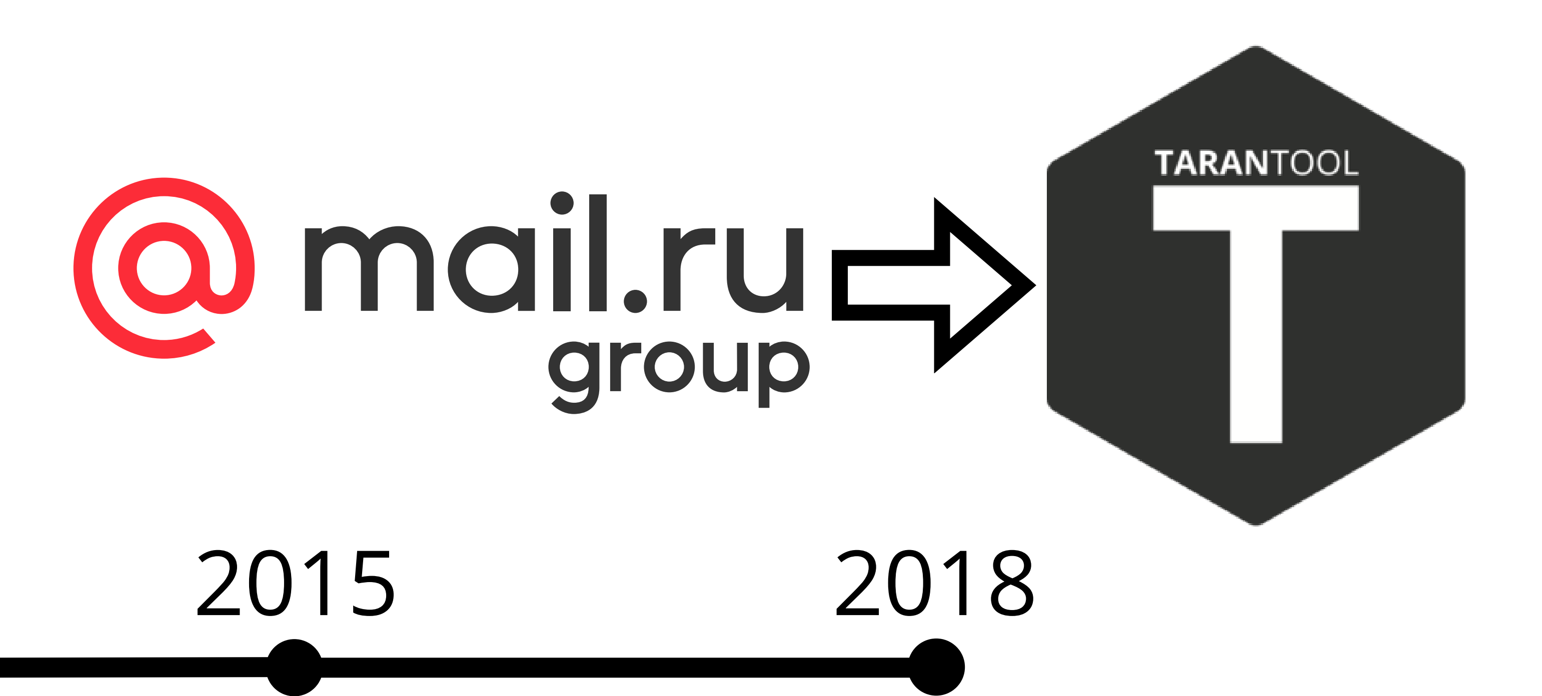
История разработки [2]



2015

Создание
плана

История разработки [2]



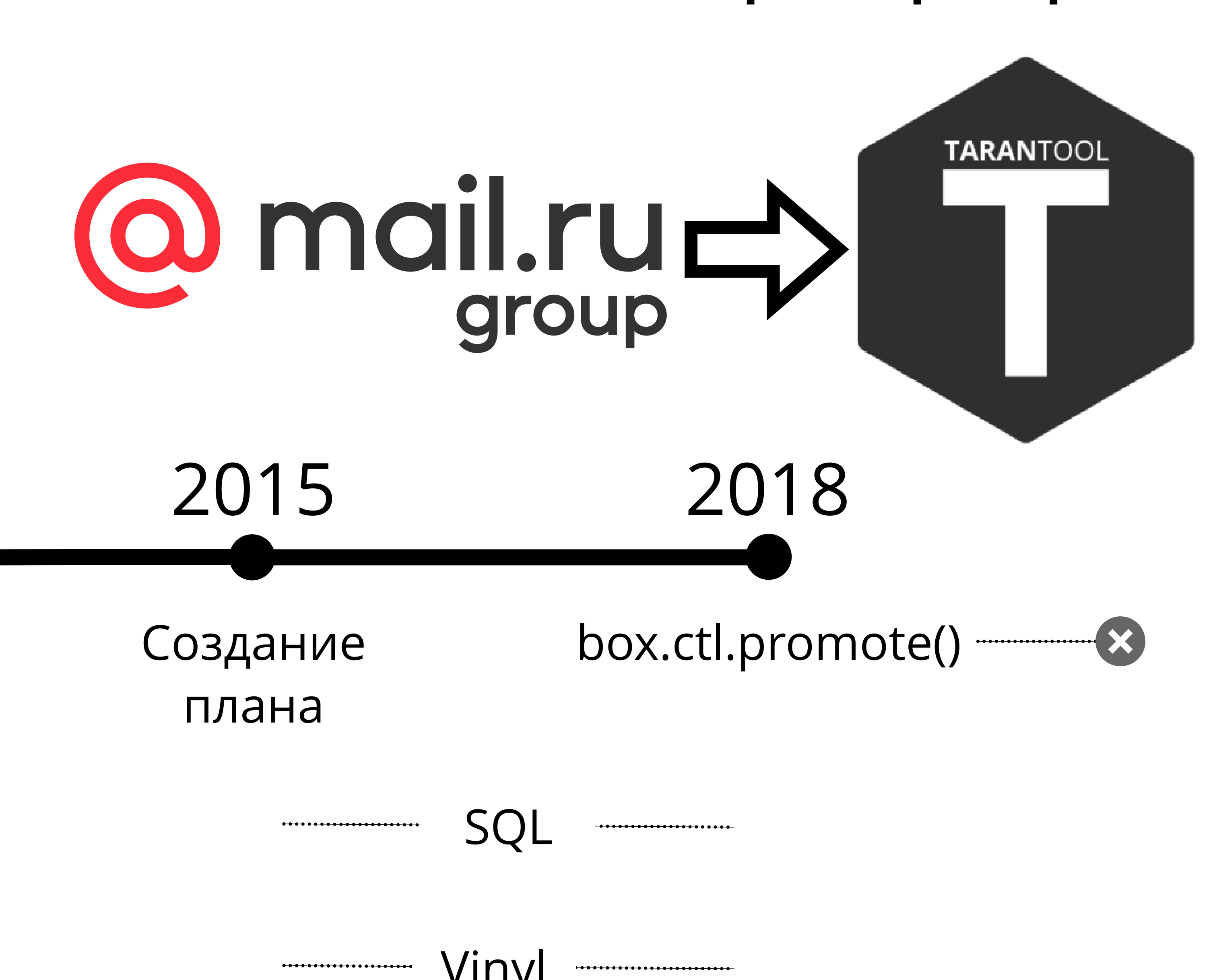
Создание
плана

SQL

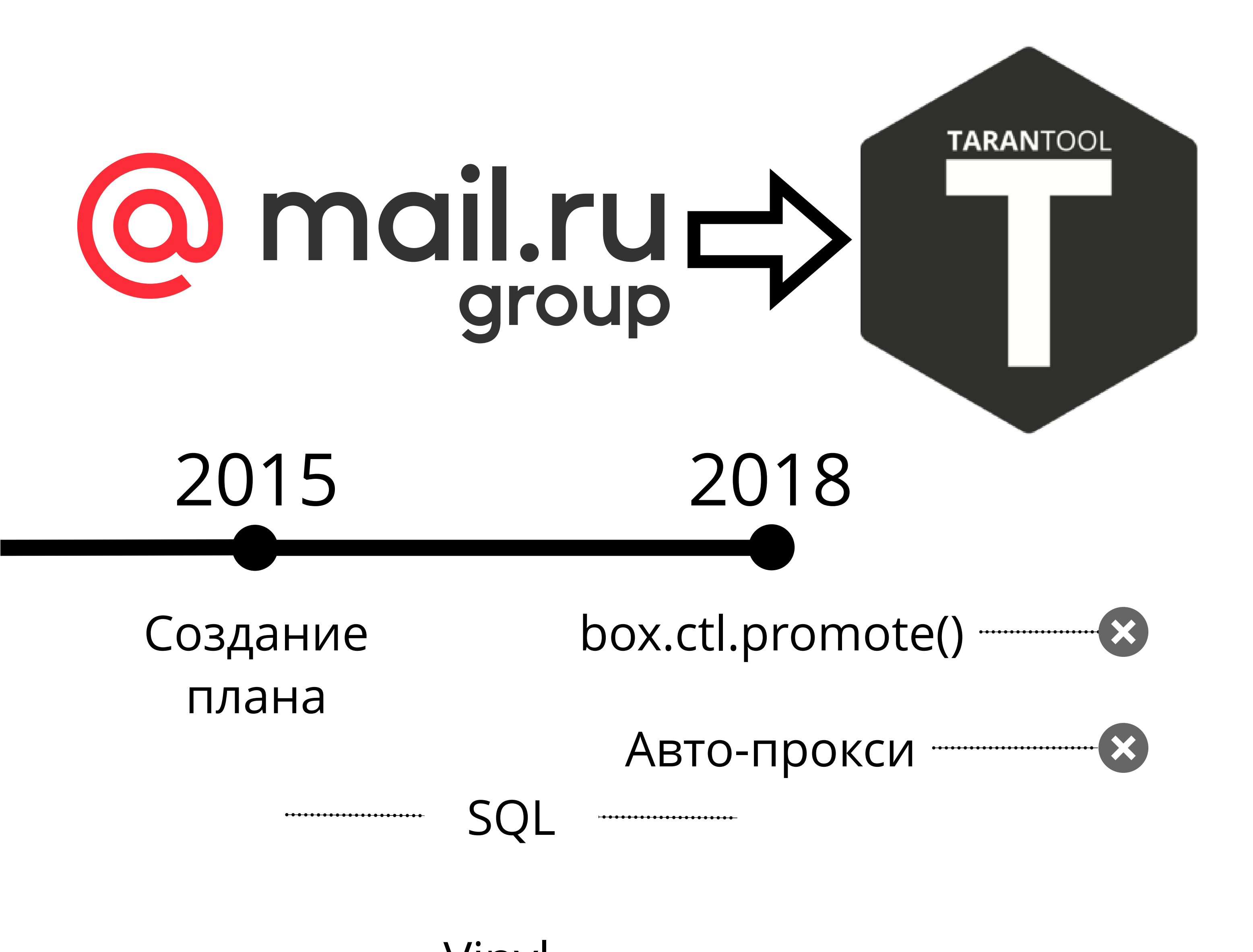
Vinyl



История разработки [2]



История разработки [2]



История разработки [2]

mail.ru group → TARANTOOL

2015 — 2018 — 2019

Создание плана

box.ctl.promote() ✗

Авто-прокси ✗

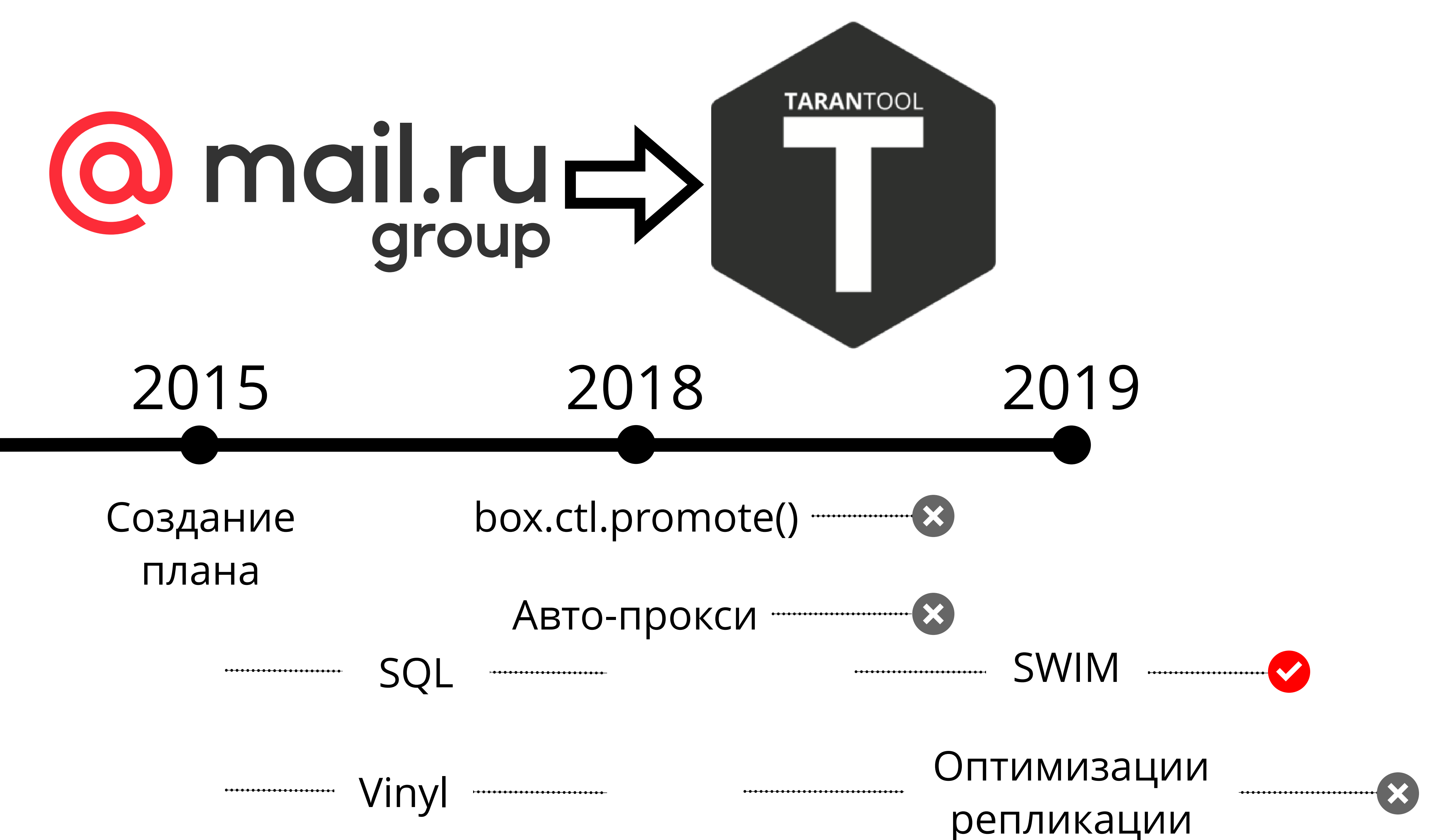
SQL

SWIM ✓

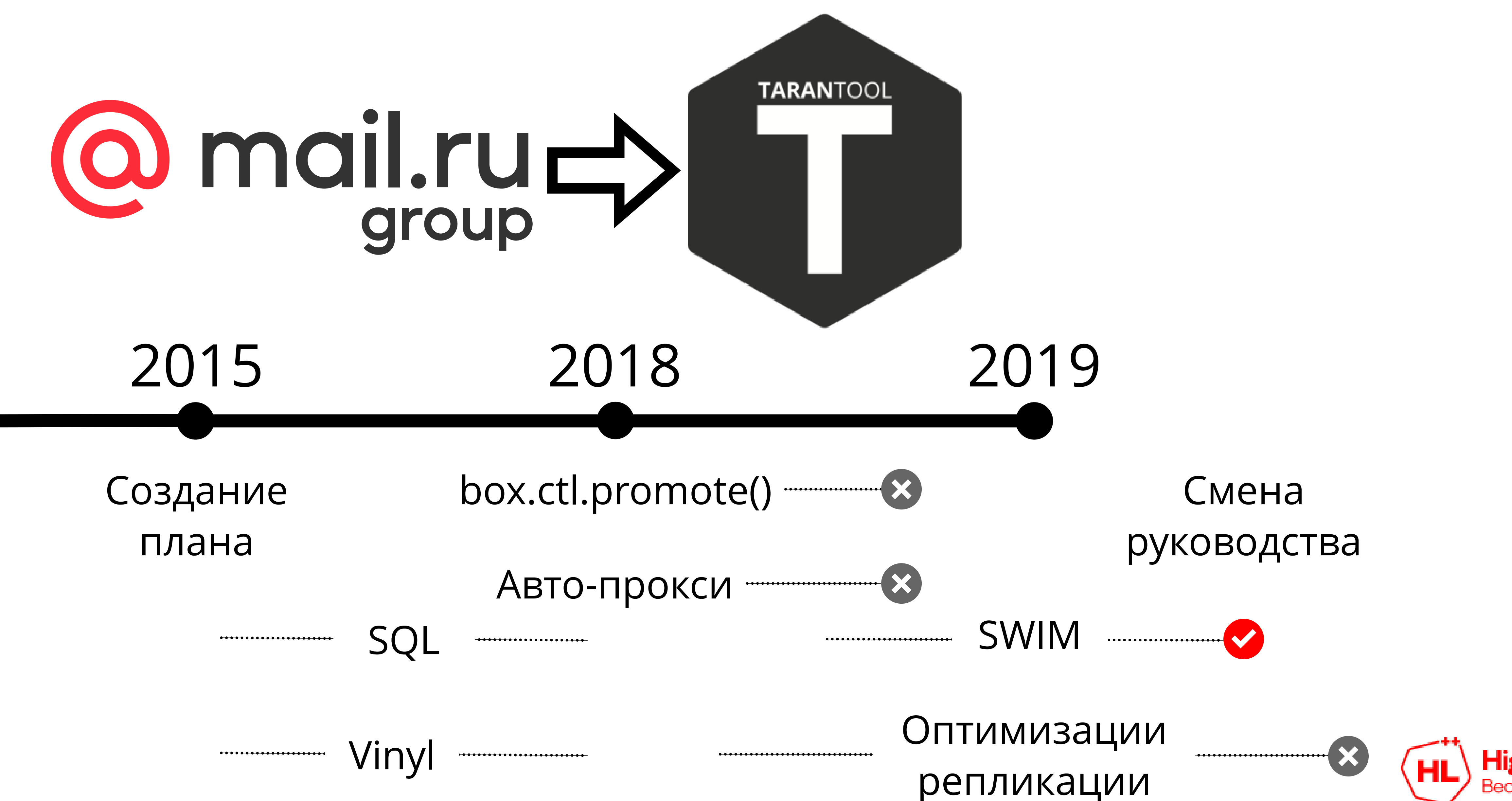
Vinyl



История разработки [2]

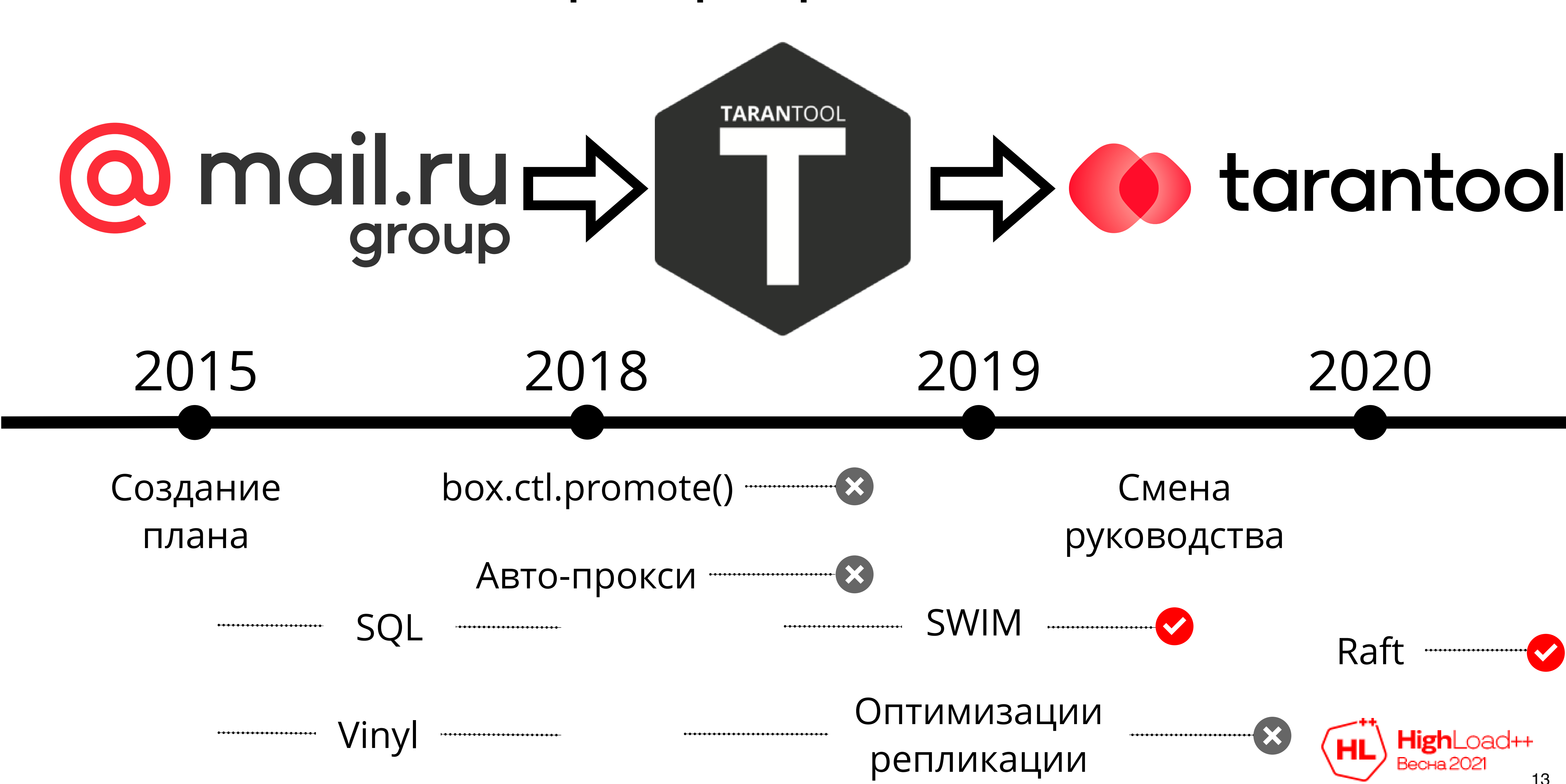


История разработки [2]



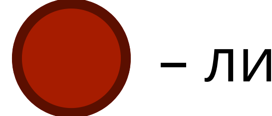
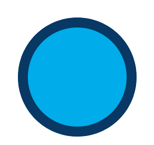
HighLoad++ Весна 2021

История разработки [2]

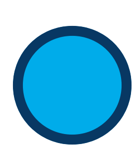
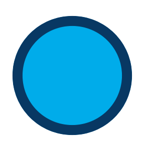
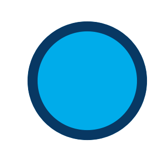
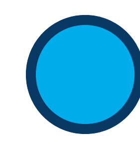
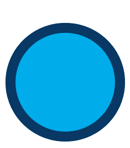


Рафт [1]: общая схема

Это Синхронная репликация и Выборы лидера

Роли:  – лидер  – реплика  – кандидат

Каждый узел изначально **реплика**

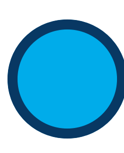






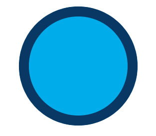
Рафт [1]: общая схема

Это Синхронная репликация и Выборы лидера

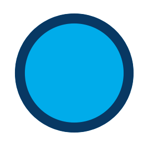
Роли:  – лидер  – реплика  – кандидат

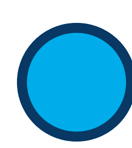
Узлы хранят персистентный **терм** – логические часы кластера


Терм: 1

Терм: 1


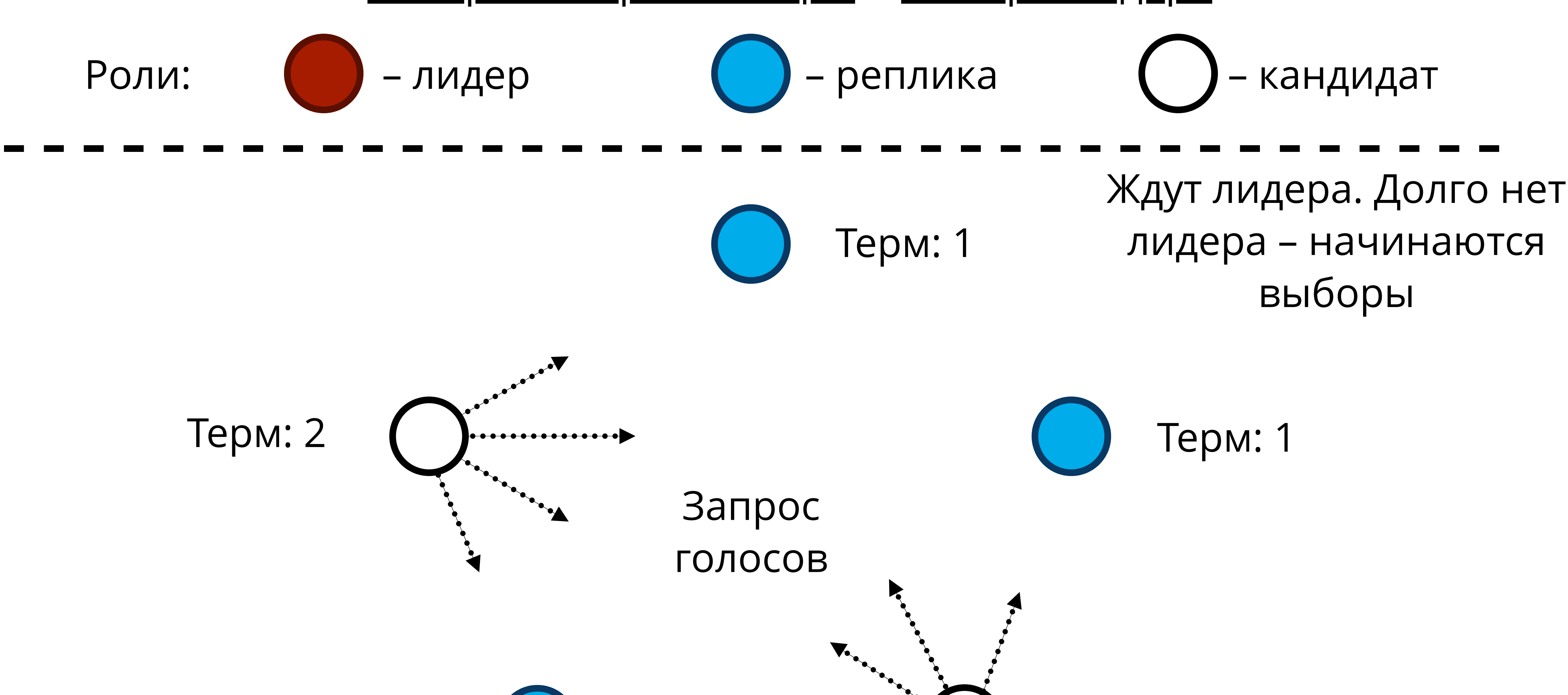
Терм: 1

Терм: 1



Терм: 1

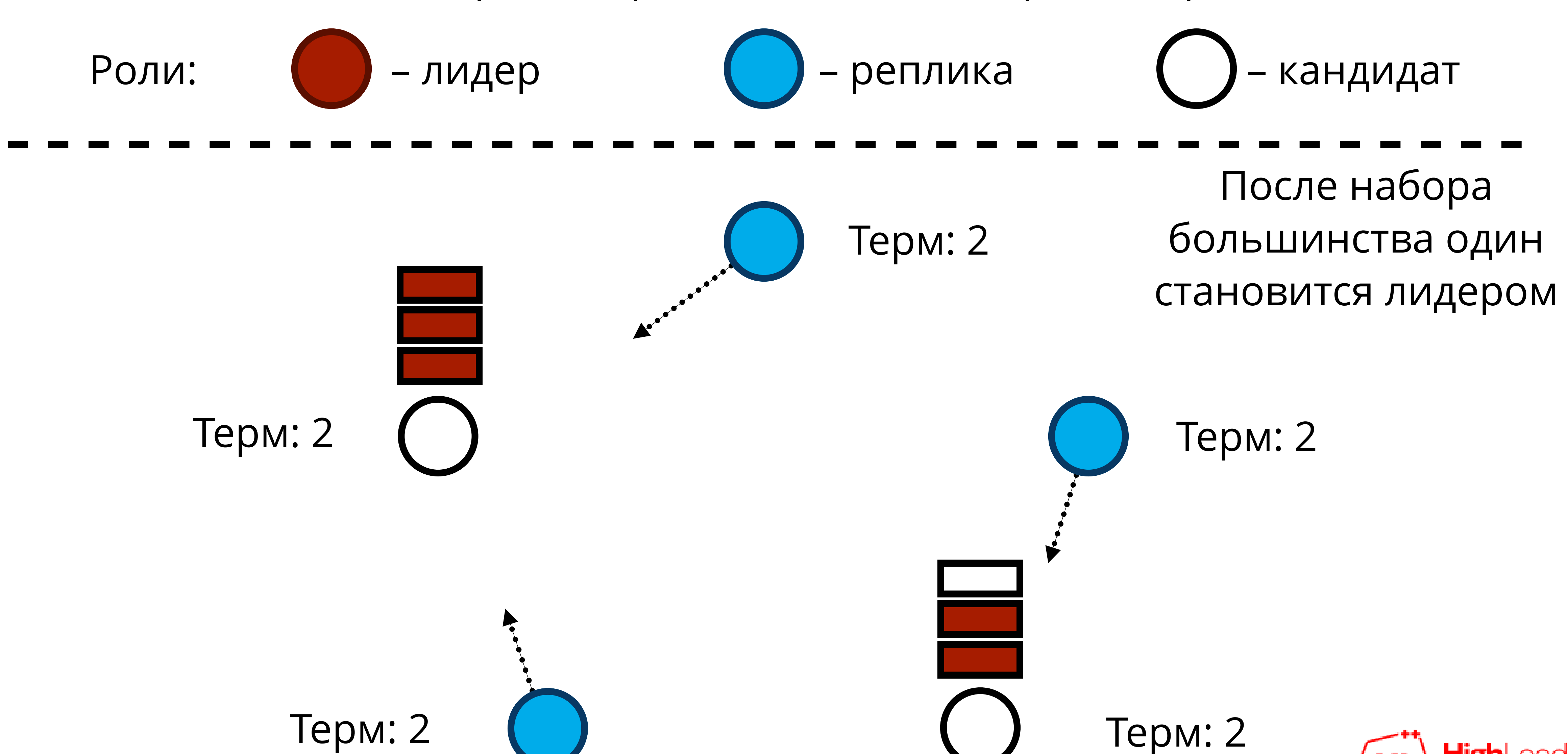
Рафт [1]: общая схема

Это Синхронная репликация и Выборы лидера



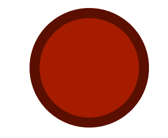
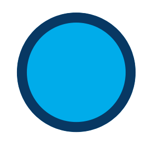
Рафт [1]: общая схема

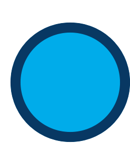
Это <u>Синхронная репликация</u> и <u>Выборы лидера</u>



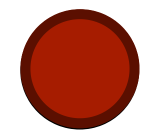
Рафт [1]: общая схема

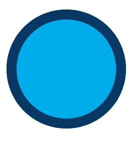
Это Синхронная репликация и Выборы лидера

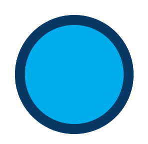
Роли:  – лидер  – реплика  – кандидат

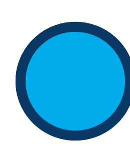
 Терм: 2

После набора большинства один становится лидером

Терм: 2 

 Терм: 2

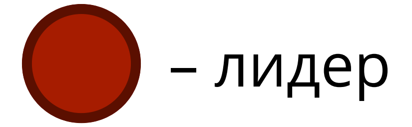
Терм: 2 

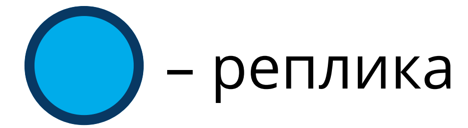
 Терм: 2

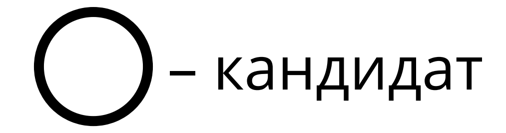
Рафт [1]: общая схема

Это Синхронная репликация и Выборы лидера





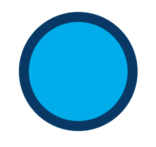




Лидер принимает все транзакции, реплицирует, и завершает коммит сразу по достижении кворума

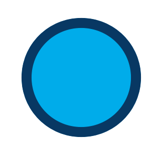


Терм: 2

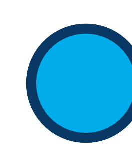


Терм: 2

Терм: 2



Терм: 2

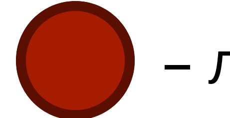
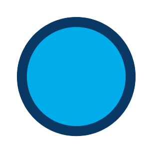
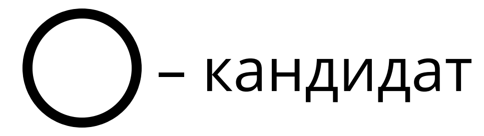


Терм: 2

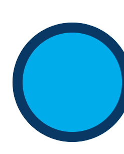
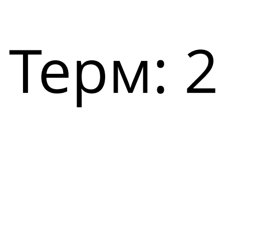
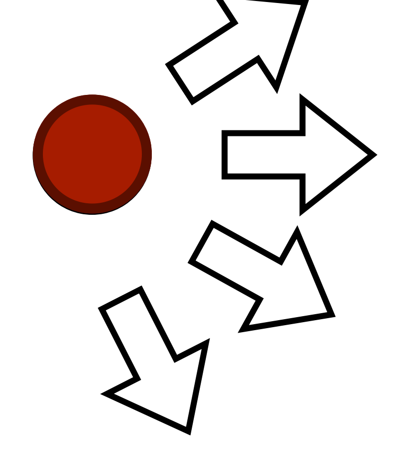
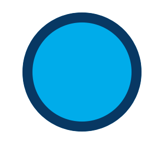
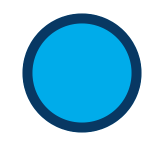
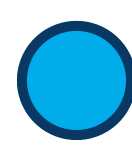


Рафт [1]: общая схема

Это Синхронная репликация и Выборы лидера

Роли: – лидер – реплика – кандидат

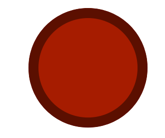







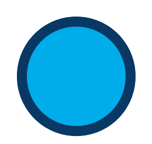
Лидер принимает все транзакции, реплицирует, и завершает коммит сразу по достижении кворума

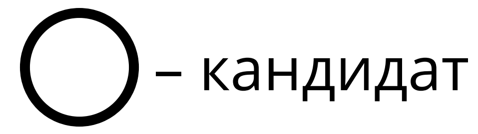
Рафт [1]: общая схема

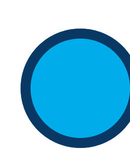
Это Синхронная репликация и Выборы лидера





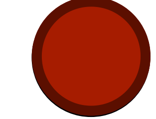


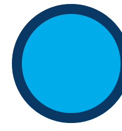




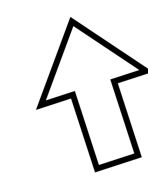
Лидер принимает все транзакции, реплицирует, и завершает коммит сразу по достижении кворума

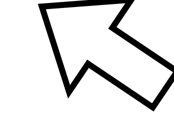


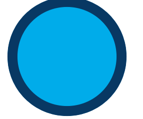


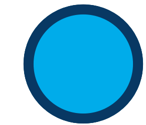


Подтверждения





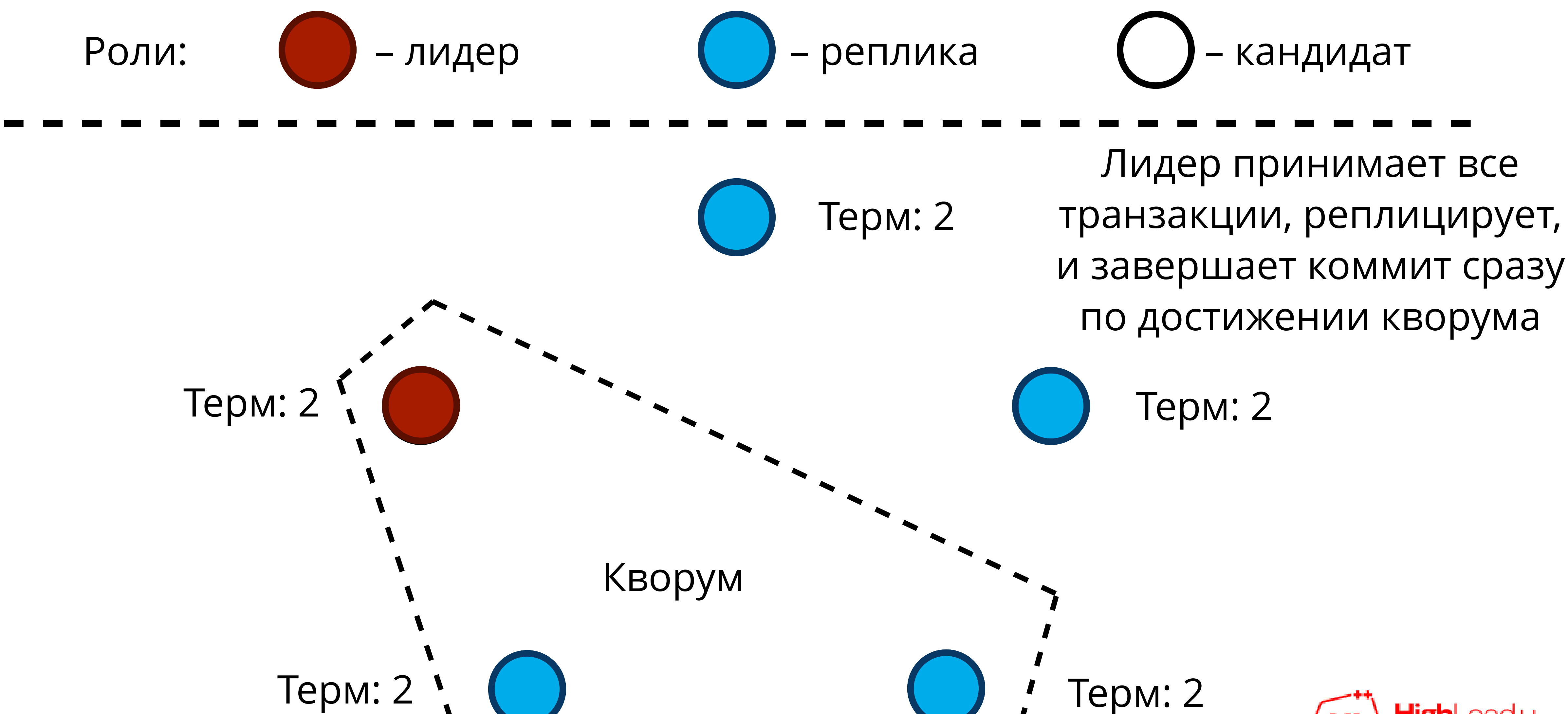






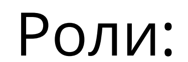
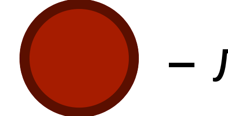
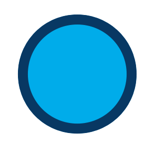
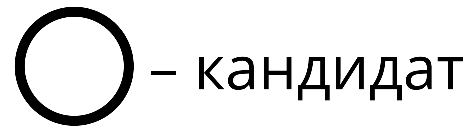
Рафт [1]: общая схема

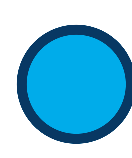
Это Синхронная репликация и Выборы лидера



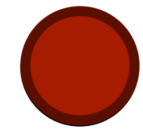
Рафт [1]: общая схема

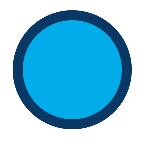
Это Синхронная репликация и Выборы лидера

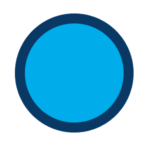

Роли:

– лидер

– реплика

– кандидат

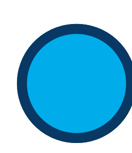

Терм: 2

Лидер принимает все транзакции, реплицирует, и завершает коммит сразу по достижении кворума


Терм: 2

Коммит


Терм: 2

Терм: 2


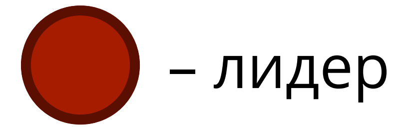

Терм: 2

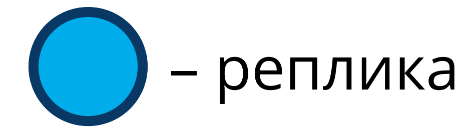


Рафт [1]: общая схема

Это Синхронная репликация и Выборы лидера

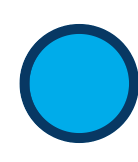






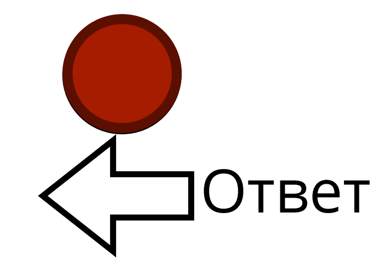


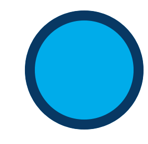
Лидер принимает все транзакции, реплицирует, и завершает коммит сразу по достижении кворума



Терм: 2

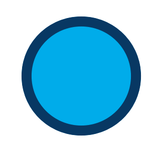


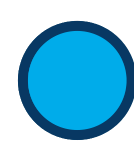




Терм: 2

Терм: 2



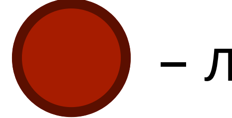


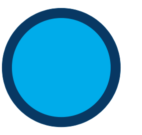
Терм: 2

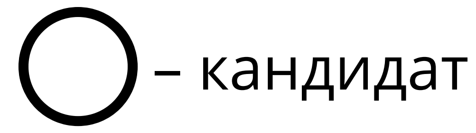
Рафт [1]: общая схема

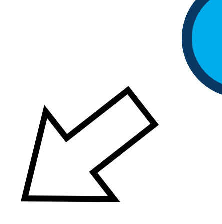
Это Синхронная репликация и Выборы лидера



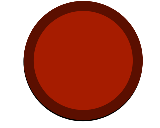




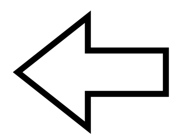
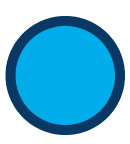


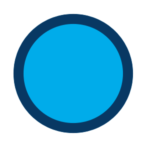


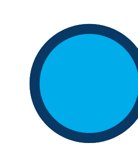
Лидер принимает все транзакции, реплицирует, и завершает коммит сразу по достижении кворума



Догонка реплик

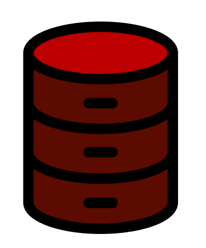
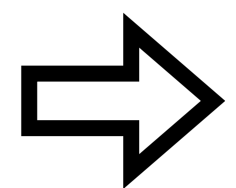


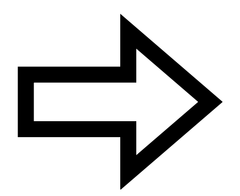
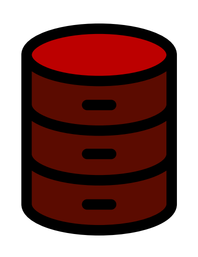
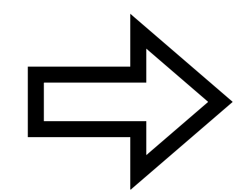




Рафт [2]: синхронная транзакция

Четыре этапа коммита транзакции


Данные в журнал лидера

Данные в журналы реплик

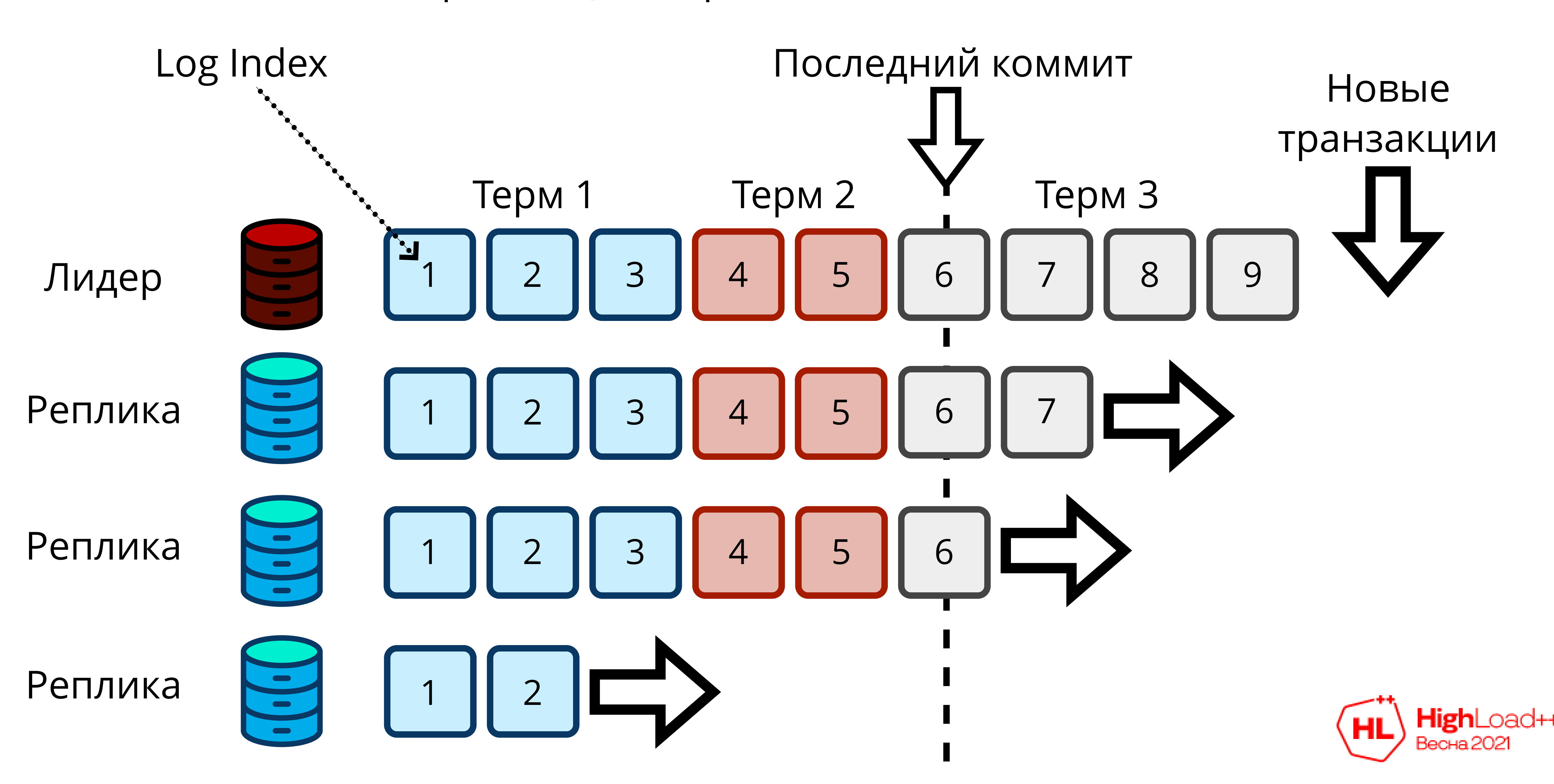
Коммит в журнал лидера и ответ пользователю

Коммит в журналы реплик

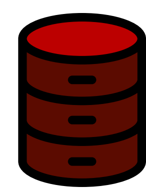
Отказ на **любом** этапе **любого** инстанса **не приводит к потере** данных после коммита, пока живо 50% + 1 инстанс

Рафт [3]: синхронный журнал

Транзакции, термы, голоса, коммиты



Рафт [4]: откат транзакций



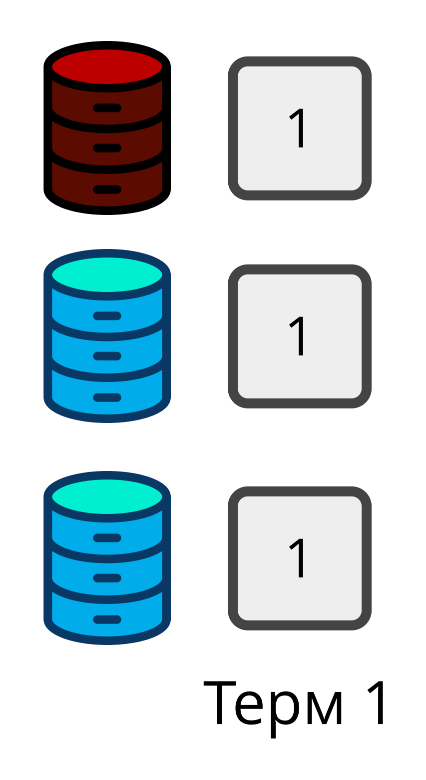




Терм 1

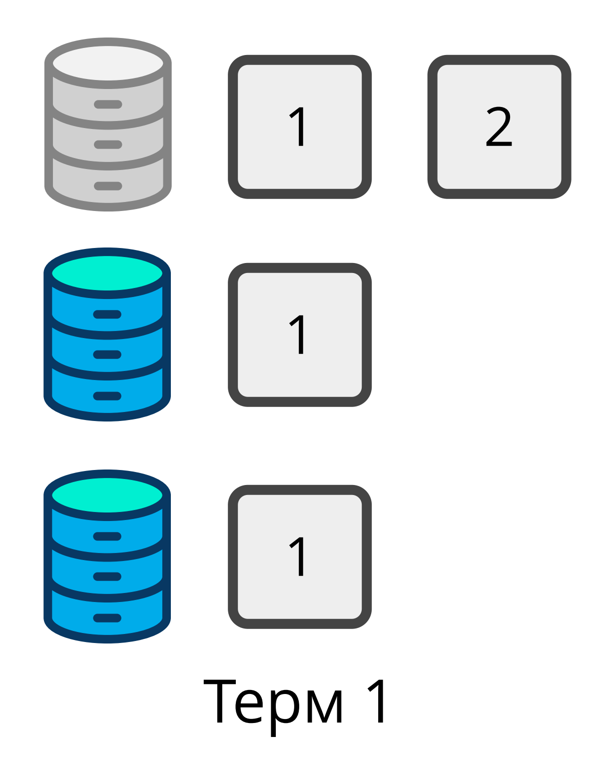
Рафт [4]: откат транзакций

Коммит первой транзакции



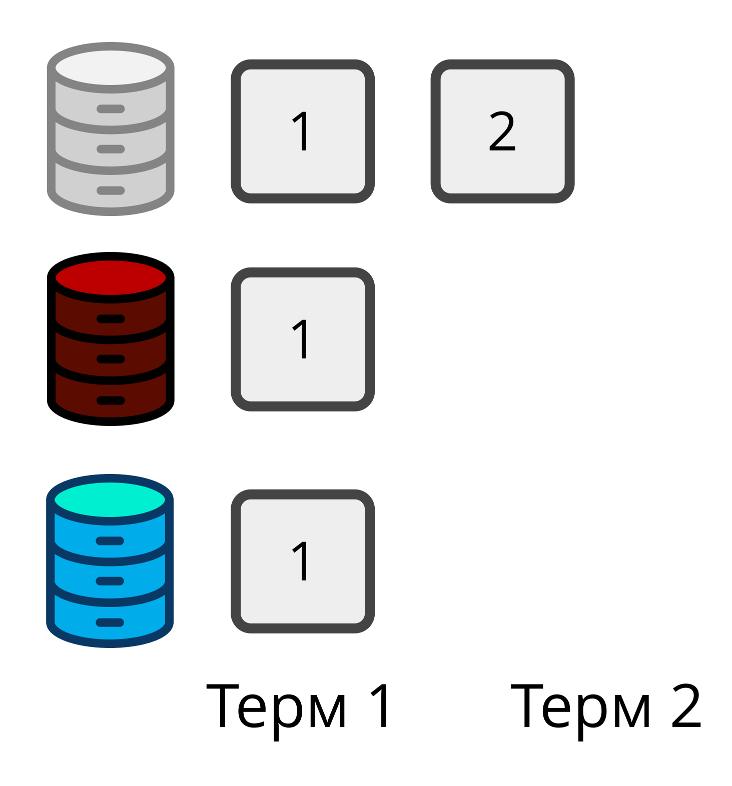
Рафт [4]: откат транзакций

Вторая в журнале, и лидер сразу офлайн



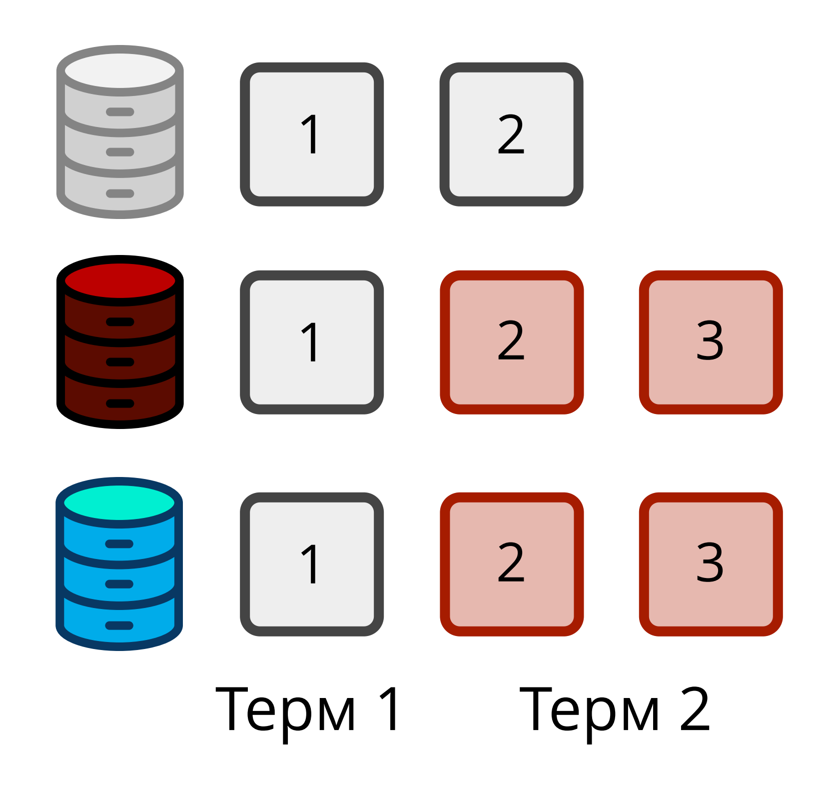
Рафт [4]: откат транзакций

Лидером выбрался другой инстанс



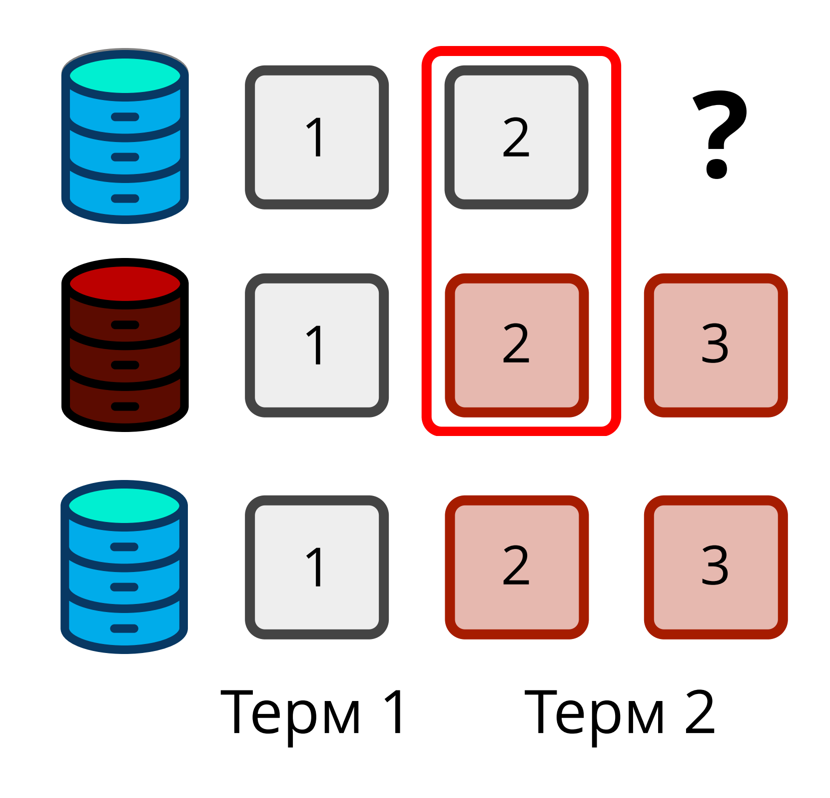
Рафт [4]: откат транзакций

Он закоммитил две других транзакции – **в новом терме**



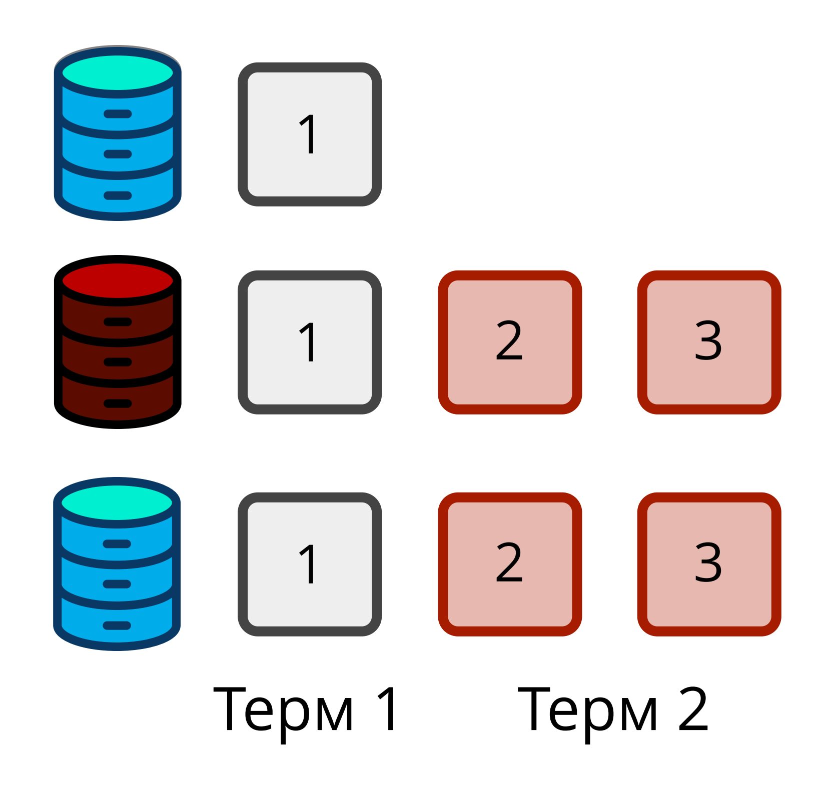
Рафт [4]: откат транзакций

Первый лидер вернулся как реплика – **конфликт!**

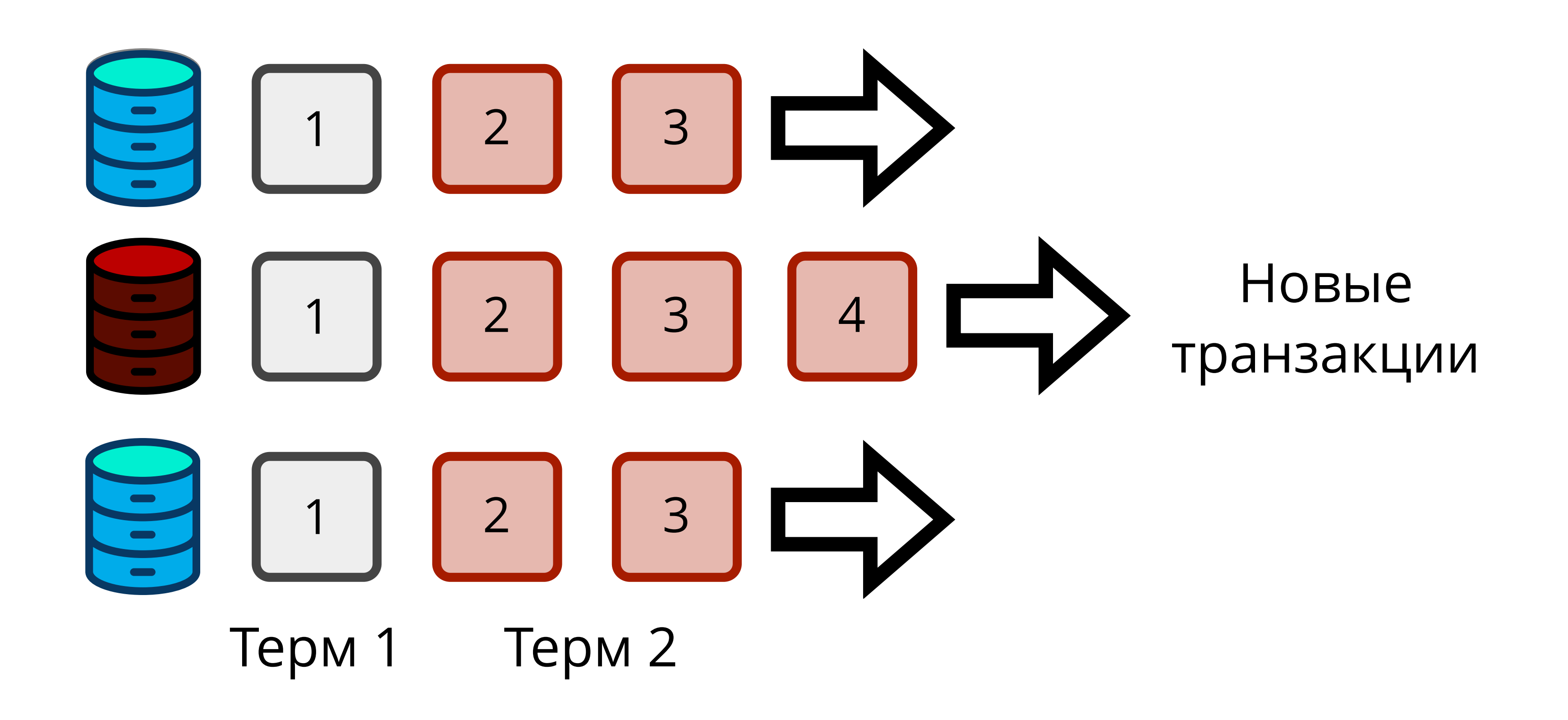


Рафт [4]: откат транзакций

Откат незакоммиченного хвоста



Рафт [4]: откат транзакций



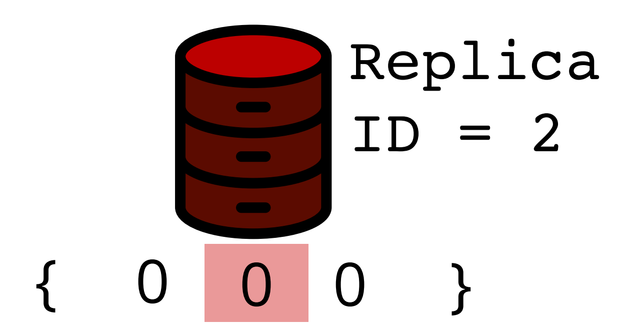
HighLoad++ Весна 2021

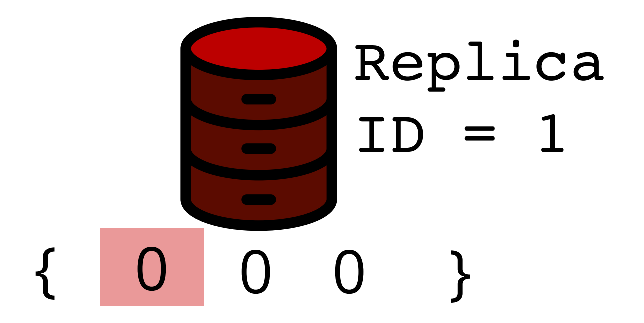
Транзакции в Tarantool

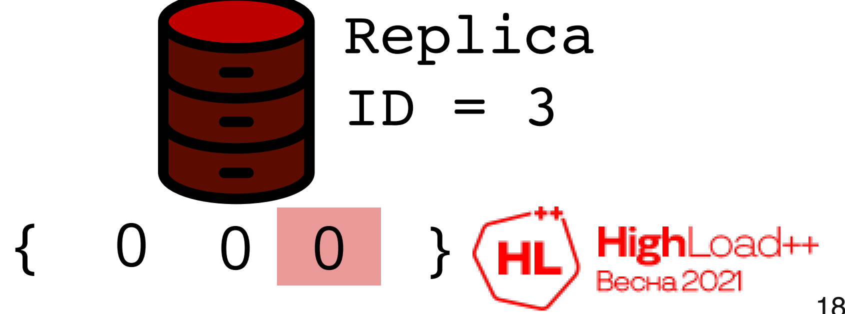
ID транзакции: **{Replica ID, LSN}**

- **Replica ID** – ID узла-создателя
- **LSN** – монотонный счетчик

VClock – пары {Replica ID, LSN}, **снимок состояния журнала**





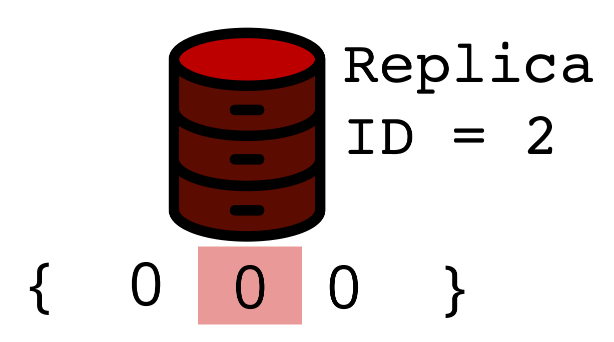


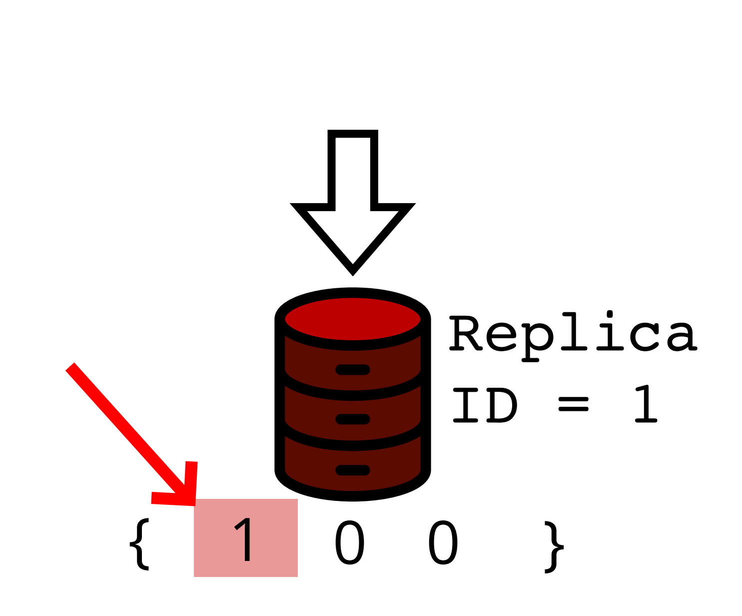
Транзакции в Tarantool

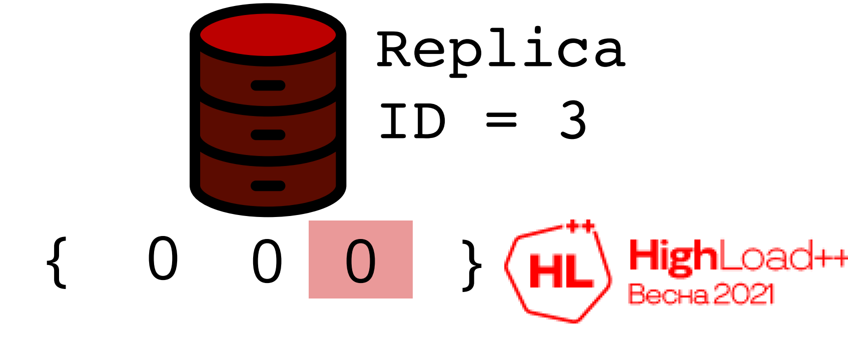
ID транзакции: **{Replica ID, LSN}**

- **Replica ID** – ID узла-создателя
- **LSN** – монотонный счетчик

VClock – пары {Replica ID, LSN}, **снимок состояния журнала**





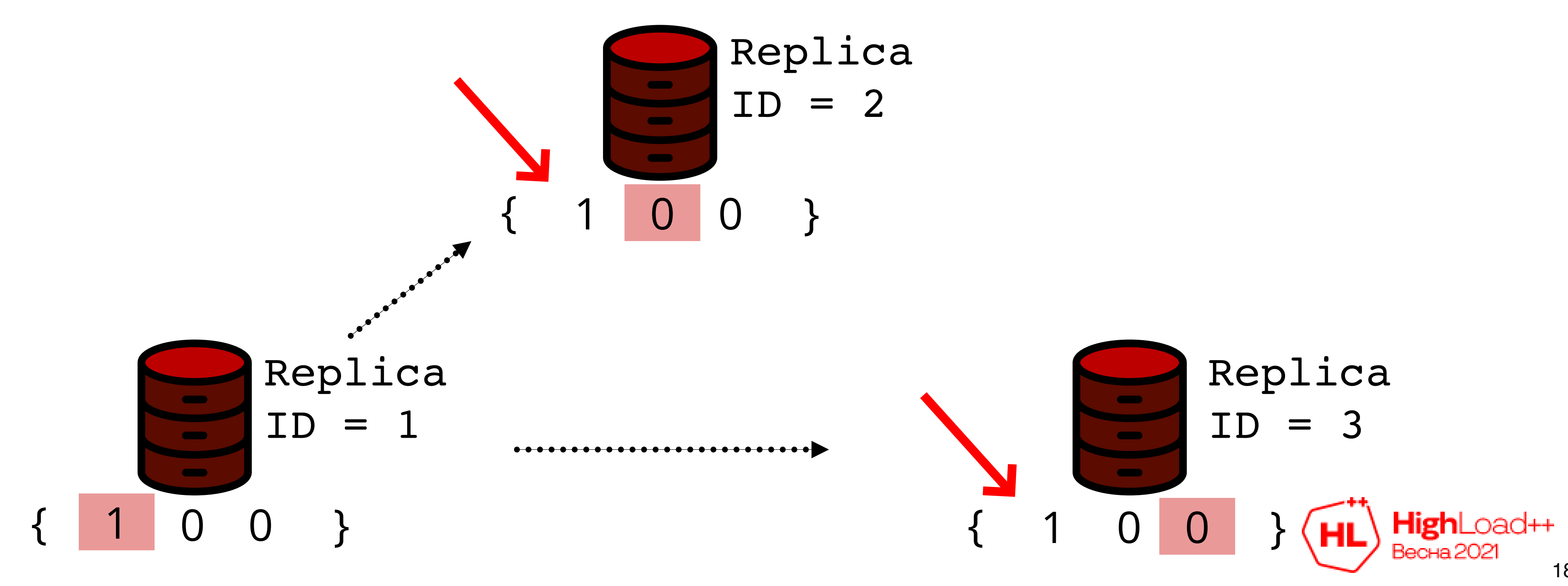


Транзакции в Tarantool

ID транзакции: **{Replica ID, LSN}**

- **Replica ID** – ID узла-создателя
- **LSN** – монотонный счетчик

VClock – пары {Replica ID, LSN}, **снимок состояния журнала**

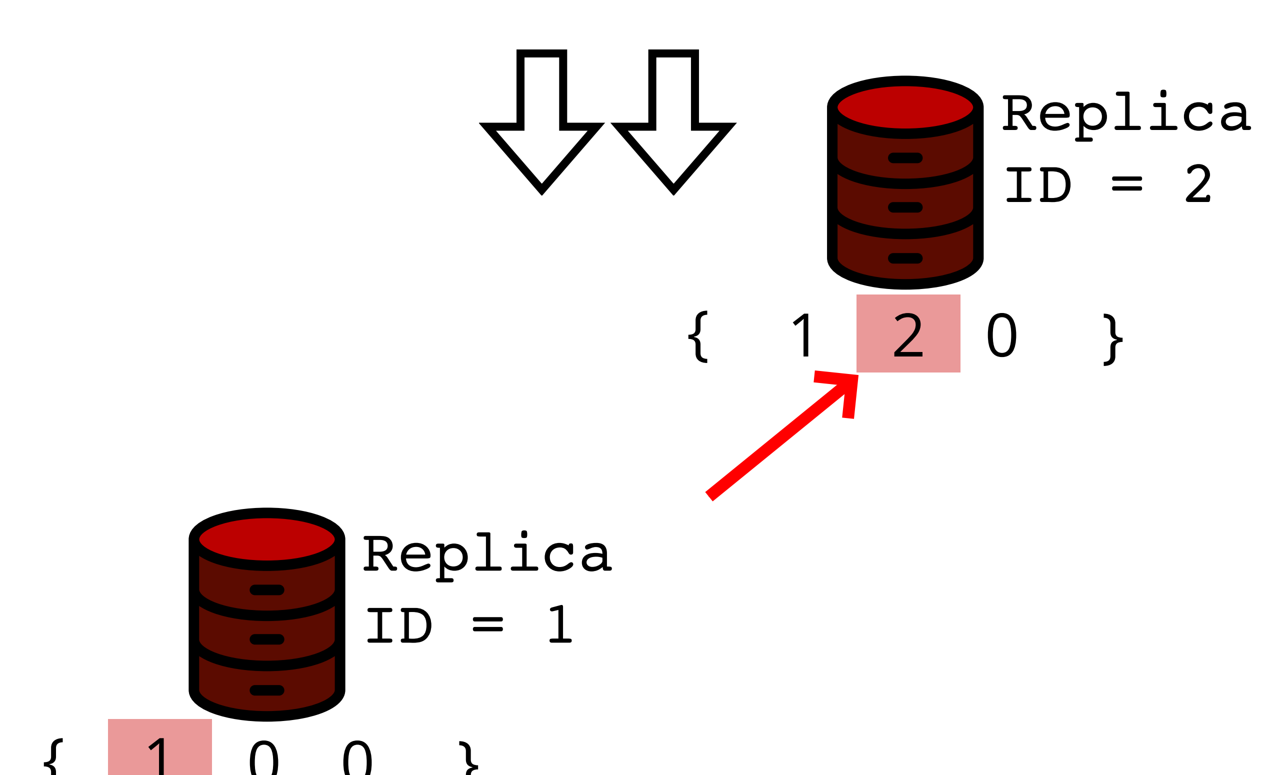


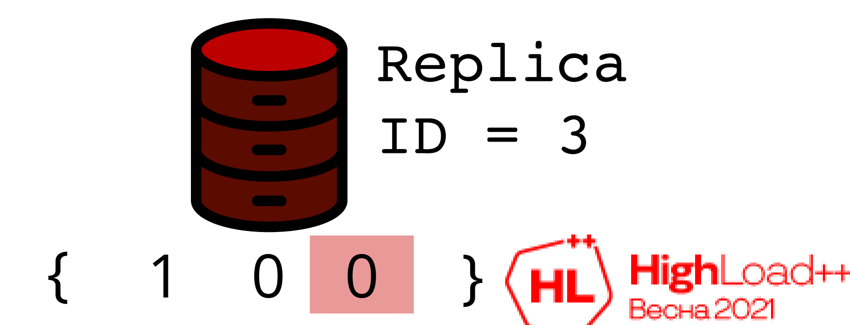
Транзакции в Tarantool

ID транзакции: **{Replica ID, LSN}**

- **Replica ID** – ID узла-создателя
- **LSN** – монотонный счетчик

VClock – пары {Replica ID, LSN}, **снимок состояния журнала**



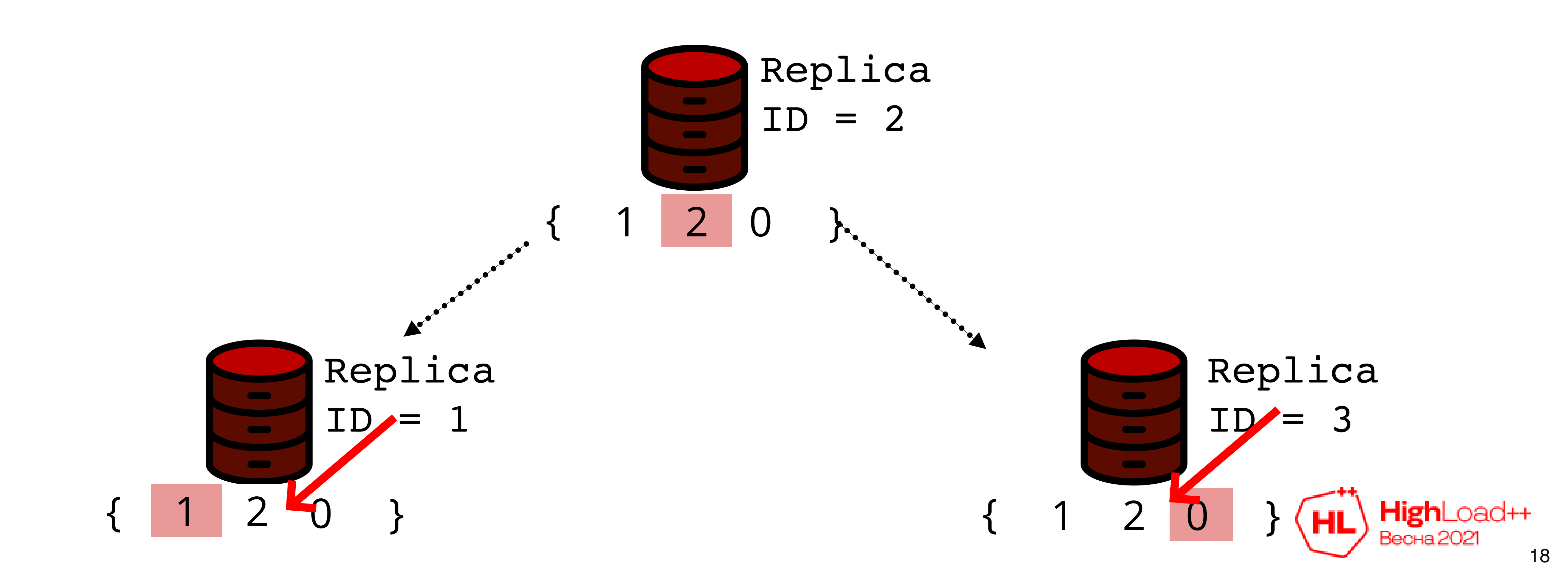


Транзакции в Tarantool

ID транзакции: **{Replica ID, LSN}**

- **Replica ID** – ID узла-создателя
- **LSN** – монотонный счетчик

VClock – пары {Replica ID, LSN}, **снимок состояния журнала**



Начало синхронной транзакции

<u>Синхронность – свойство транзакции</u>

```
sync = box.schema.create_space(
    ‘stest’, {is_sync = true}
):create_index(‘pk’)

sync:replace{1}

box.begin()
sync:replace{2}
sync:replace{3}
box.commit()
```

Это синхронные транзакции

Начало синхронной транзакции

Синхронность – свойство транзакции

```
sync = box.schema.create_space(
    'stest', {is_sync = true}
):create_index('pk')

sync:replace{1}

box.begin()
sync:replace{2}
sync:replace{3}
box.commit()
```

Один синхронный спейс – вся транзакция тоже

```
async = box.schema.create_space(
    'atest', {is_sync = false}
):create_index('pk')

box.begin()
sync:replace{5}
async:replace{6}
box.commit()
```

Это синхронные транзакции

Начало синхронной транзакции

<u>Синхронность – свойство транзакции</u>

```
sync = box.schema.create_space(
    ‘stest’, {is_sync = true}
):create_index(‘pk’)

sync:replace{1}

box.begin()
sync:replace{2}
sync:replace{3}
box.commit()
```

<u>Один синхронный спейс – вся транзакция тоже</u>

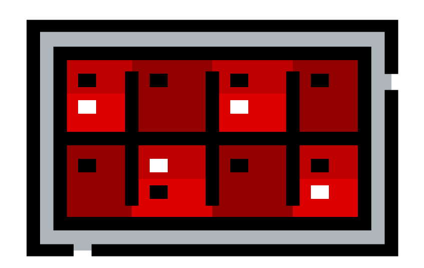
```
async = box.schema.create_space(
    ‘atest’, {is_sync = false}
):create_index(‘pk’)

box.begin()
sync:replace{5}
async:replace{6}
box.commit()
```

Это синхронные транзакции

Коммит начинается с записи в журнал мастера

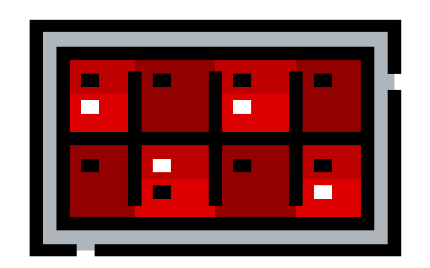
Ожидание кворума [1]: лимб



После журнала транзакция попадает в **лимб**

HighLoad++ Весна 2021

Ожидание кворума [1]: лимб



После журнала транзакция попадает в **лимб**

Лимб – это очередь синхронных транзакций

Новые транзакции

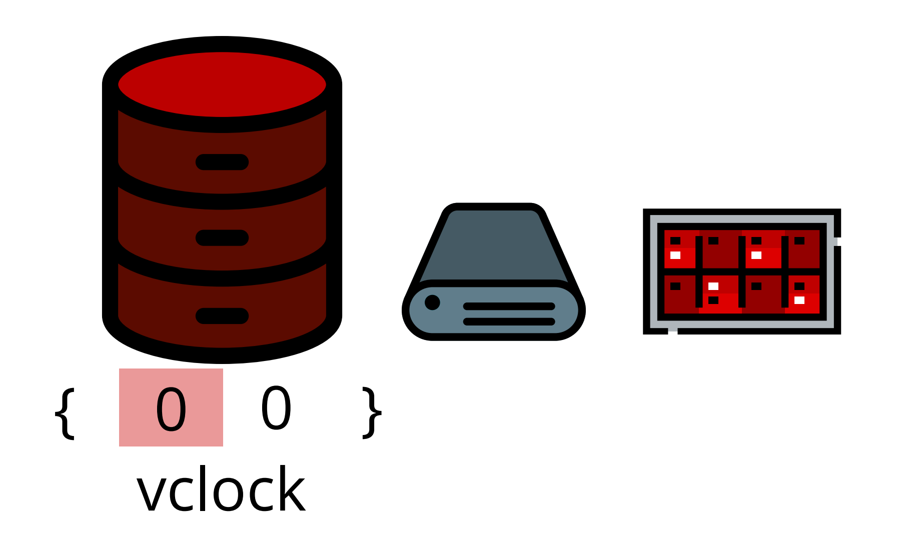
Кворум

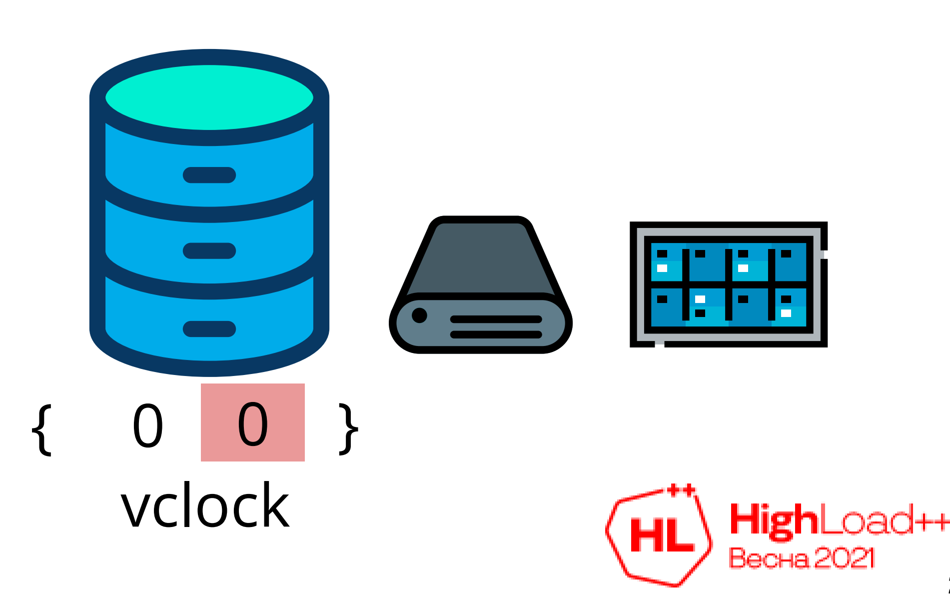
5 → 7 → 10 ... → LSN

Закоммиченные транзакции

Ожидание кворума [2]: репликация

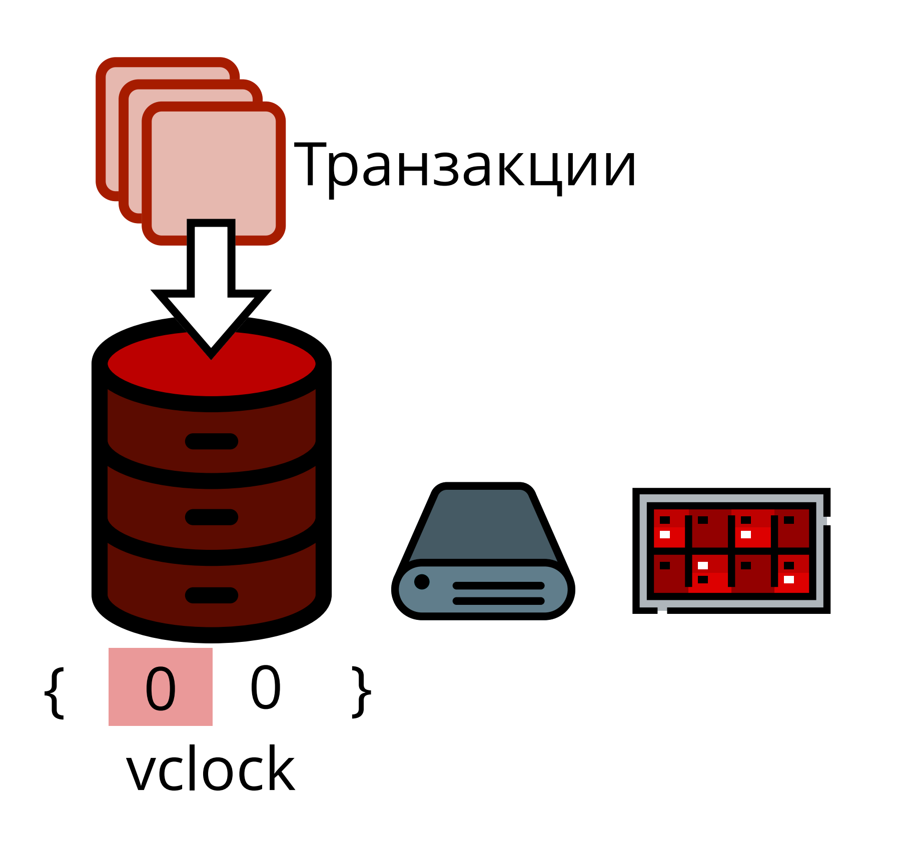
Синхронные транзакции надо доставить
на реплики и собрать подтверждения

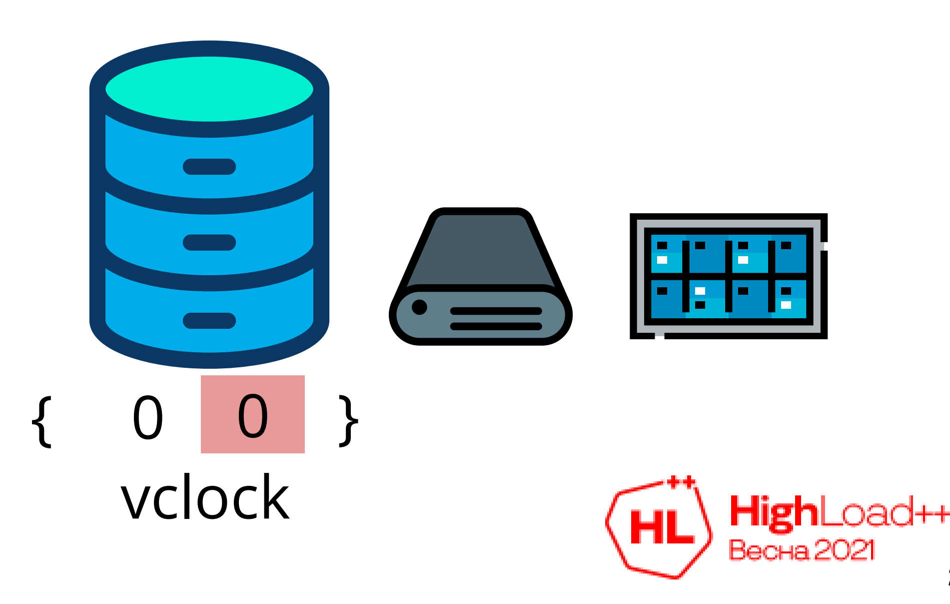




Ожидание кворума [2]: репликация

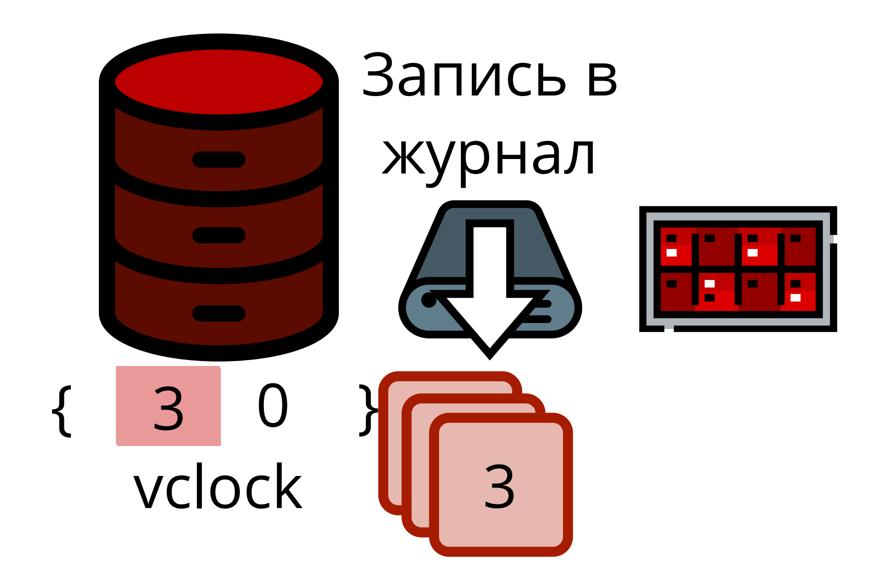
Мастер делает транзакции

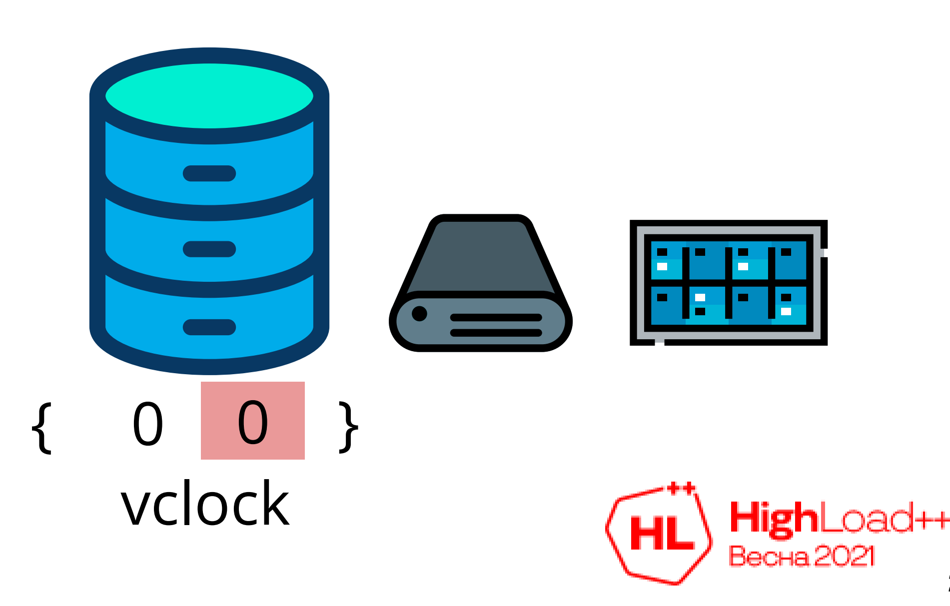




Ожидание кворума [2]: репликация

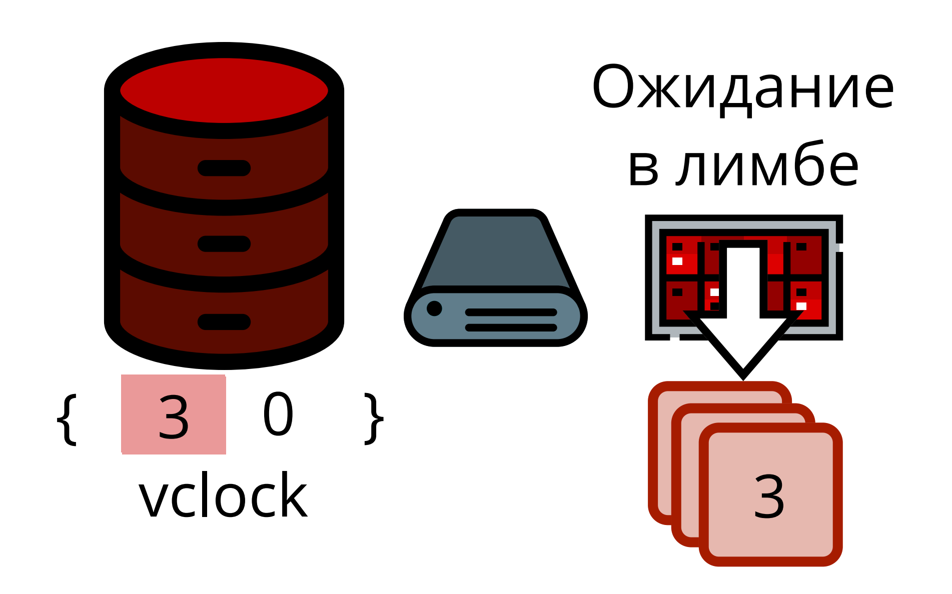
Пишет в свой журнал

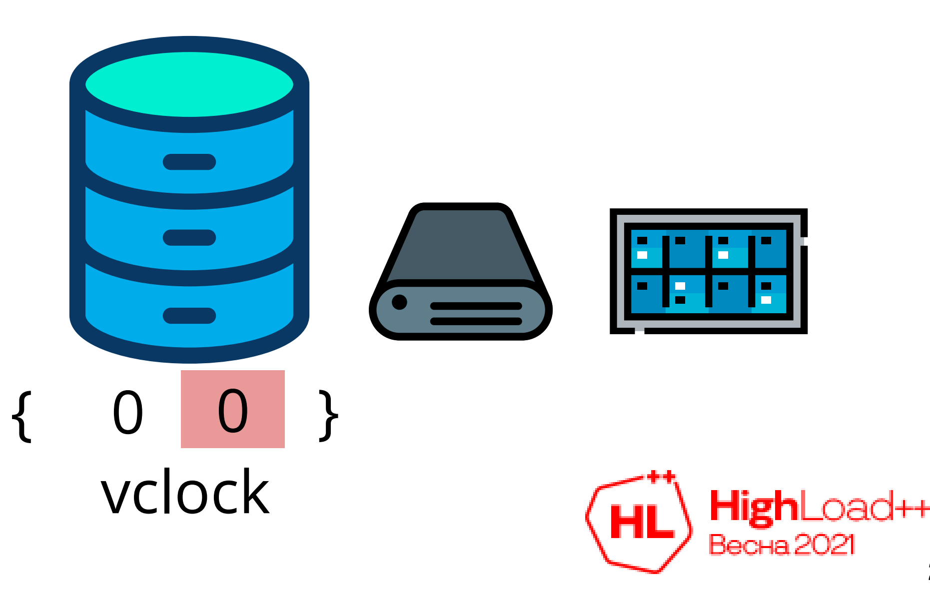




Ожидание кворума [2]: репликация

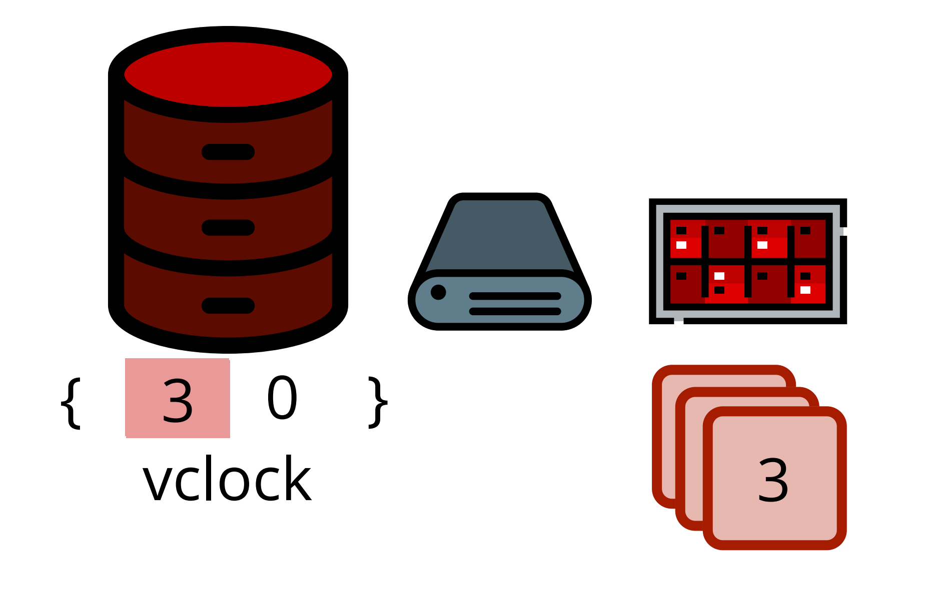
Они попадают в лимб



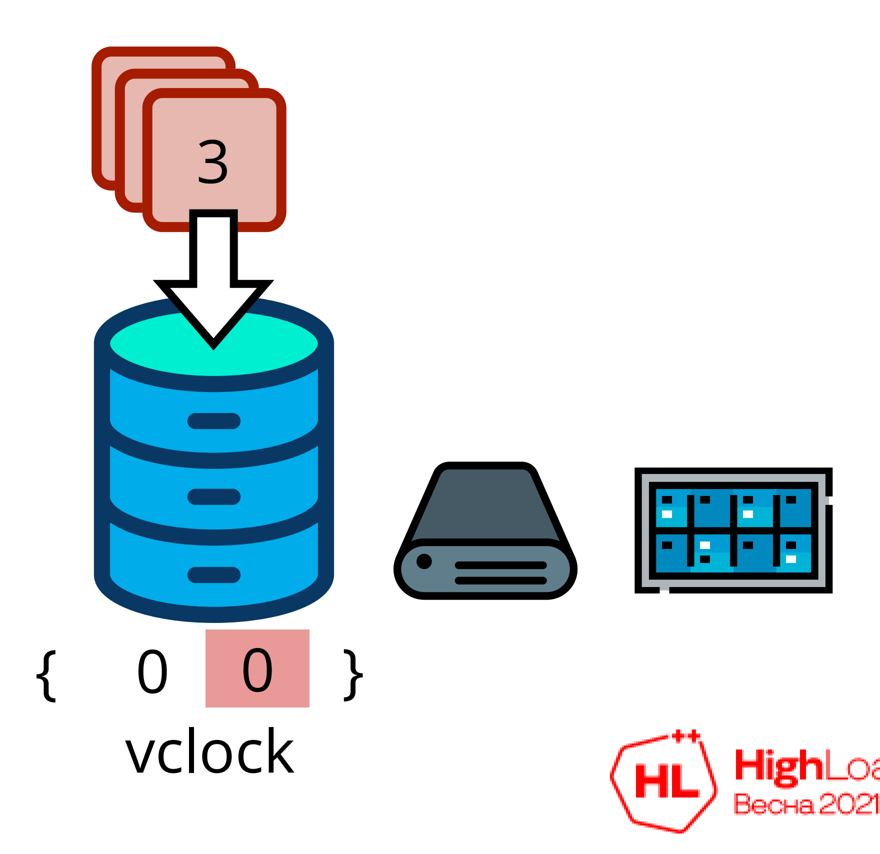


Ожидание кворума [2]: репликация

Происходит репликация на другие инстансы

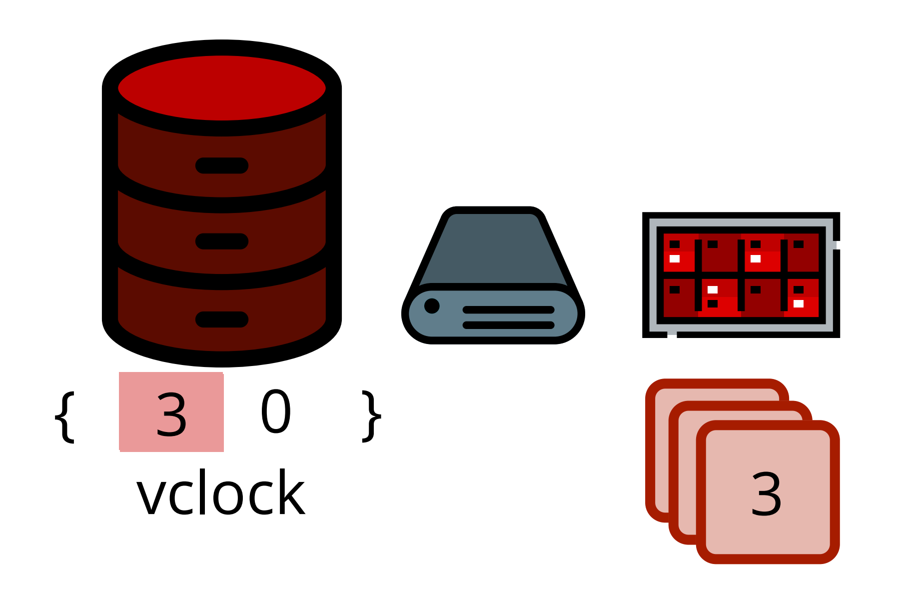


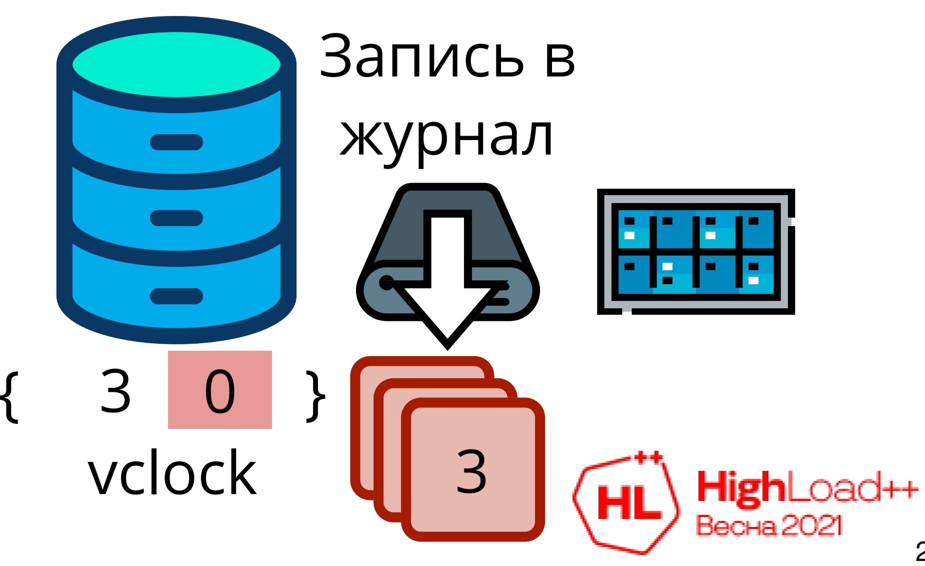
Репликация



Ожидание кворума [2]: репликация

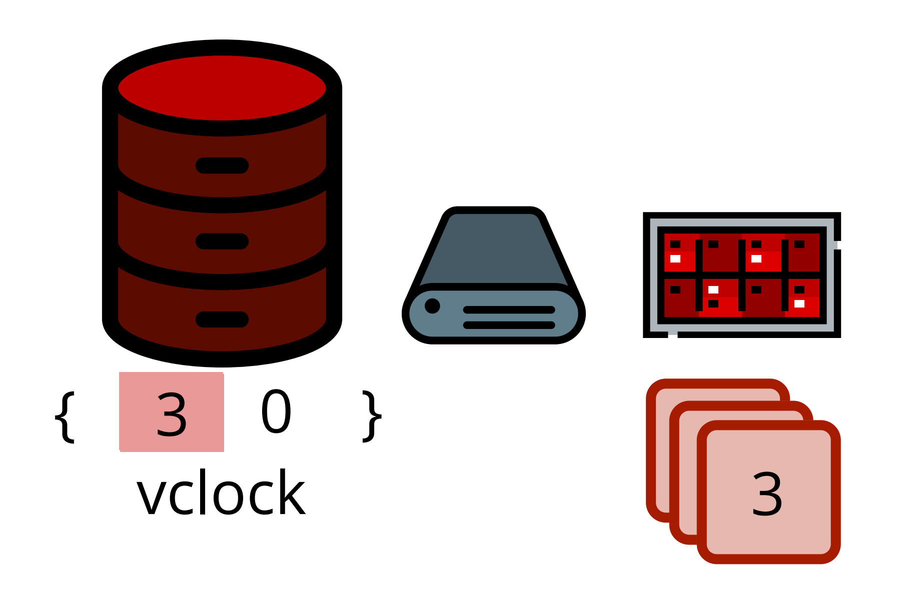
Реплика пишет эти же транзакции в свой журнал

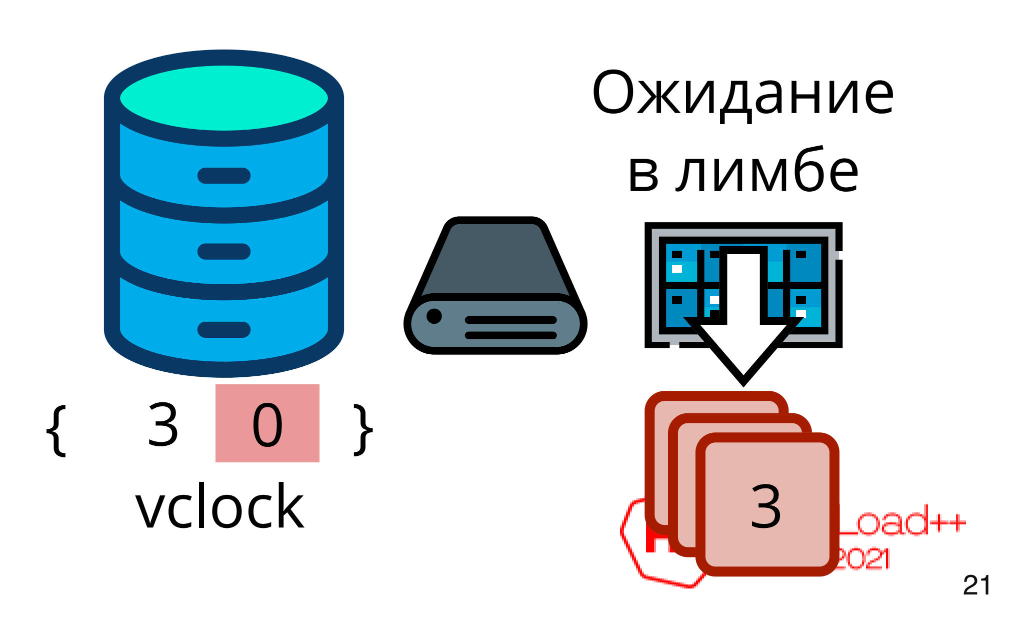




Ожидание кворума [2]: репликация

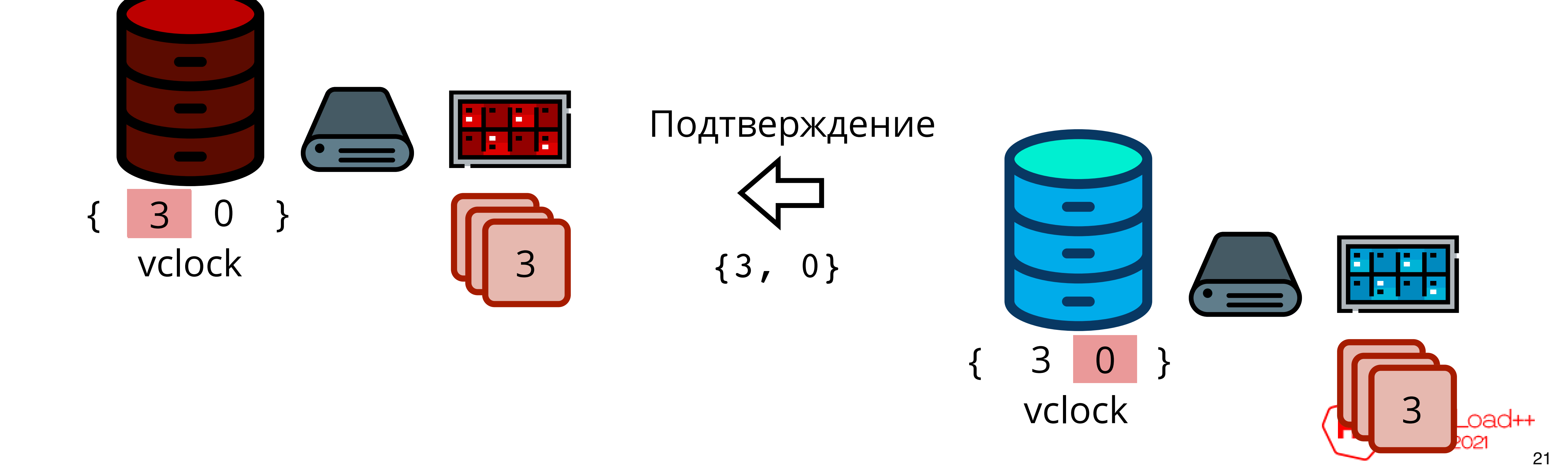
И они попадают снова в лимб, но на реплике



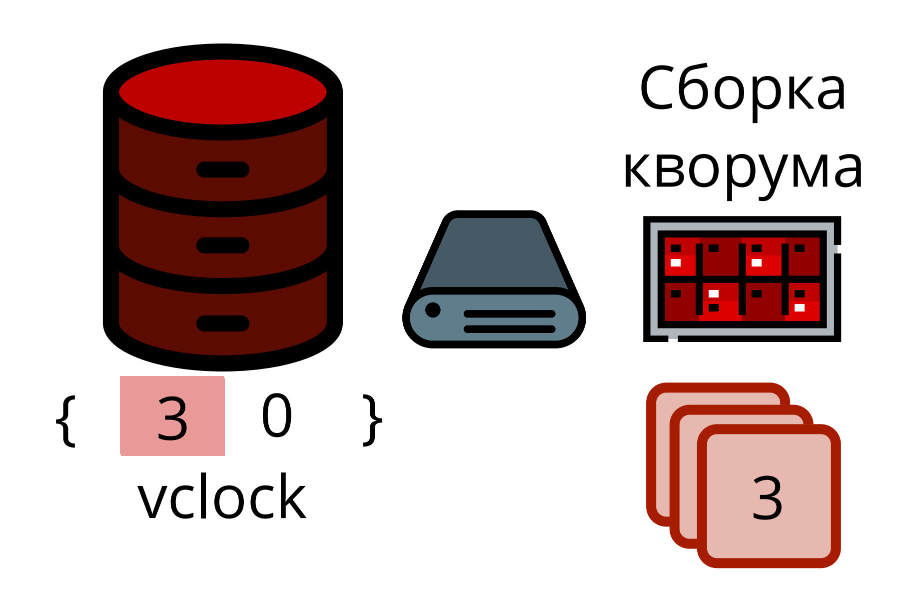


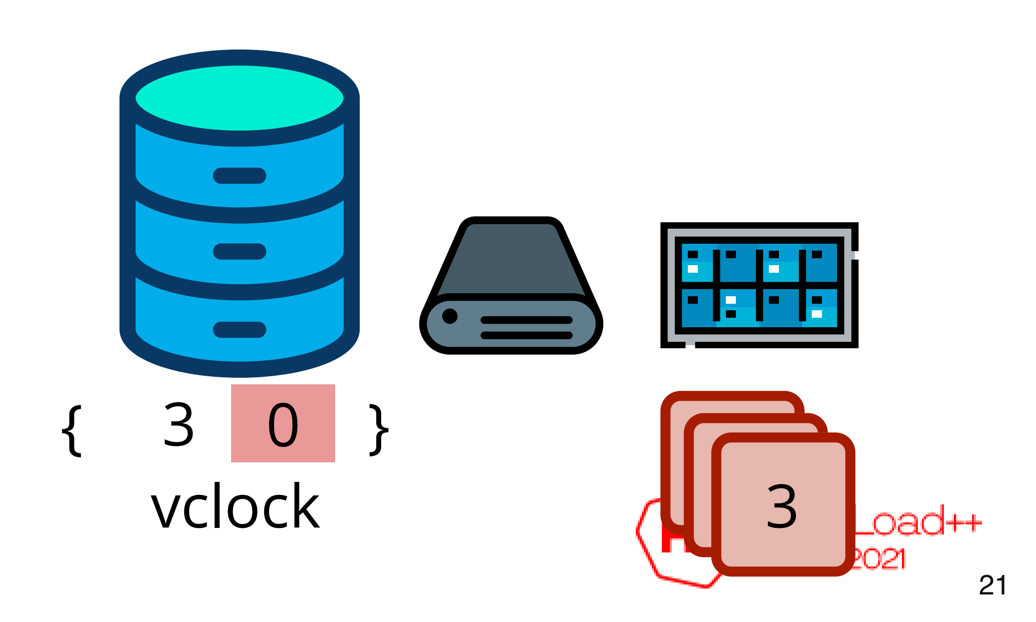
Ожидание кворума [2]: репликация

Реплика посылает лидеру подтверждение



Ожидание кворума [2]: репликация

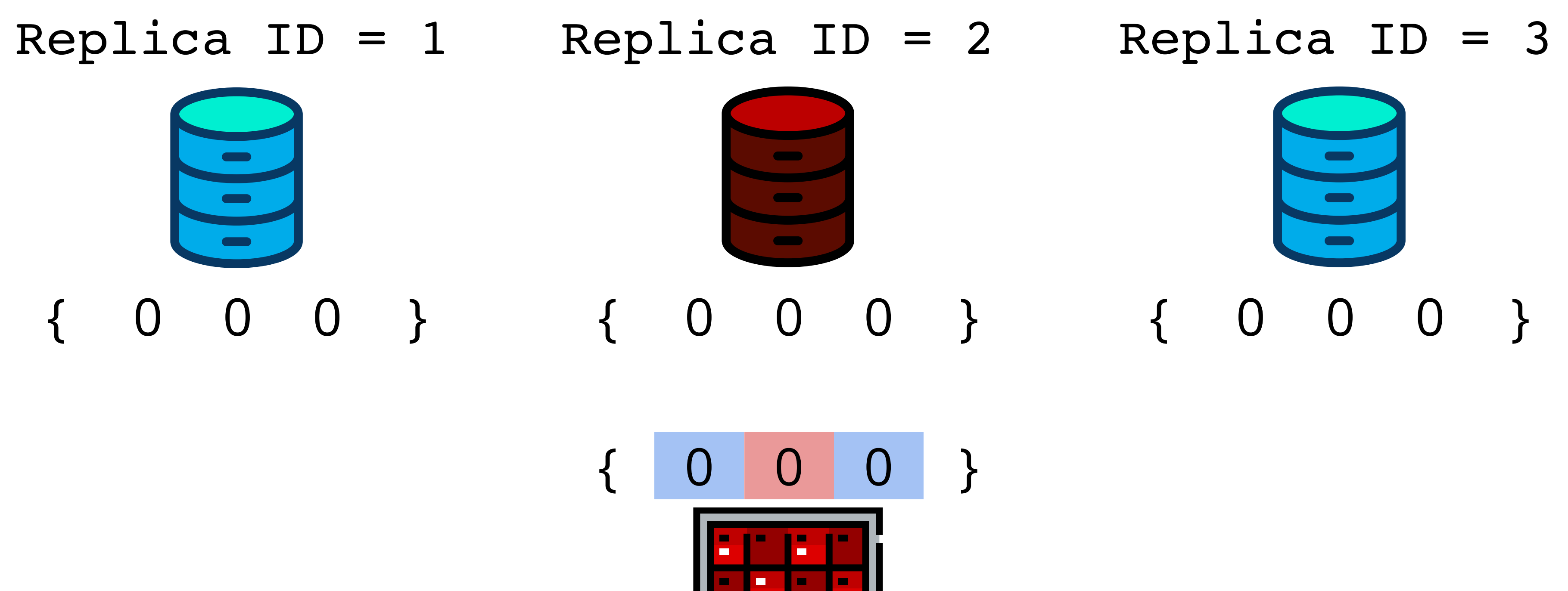




Ожидание кворума [3]: сборка

Лимб использует особый vclock

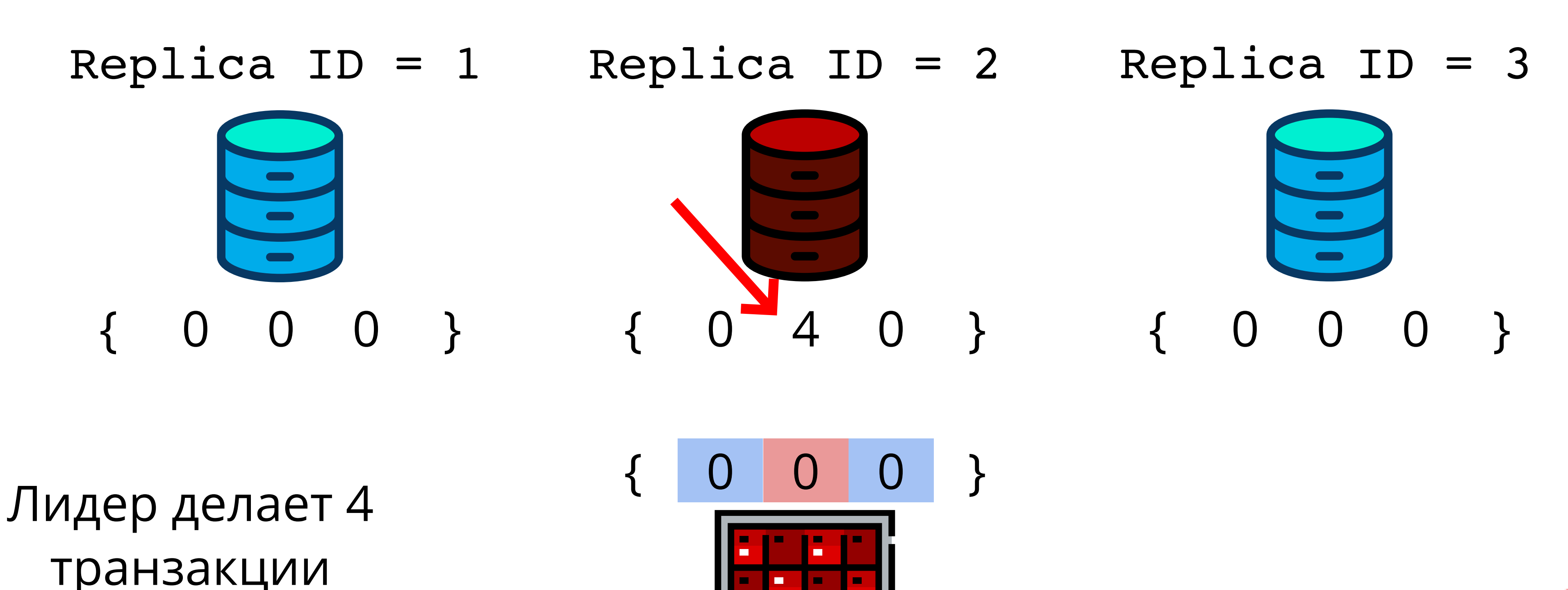
- **Replica ID** – ID реплики
- **LSN** – последний LSN **лидера,** подтвержденный этой **репликой**



Ожидание кворума [3]: сборка

Лимб использует особый vclock

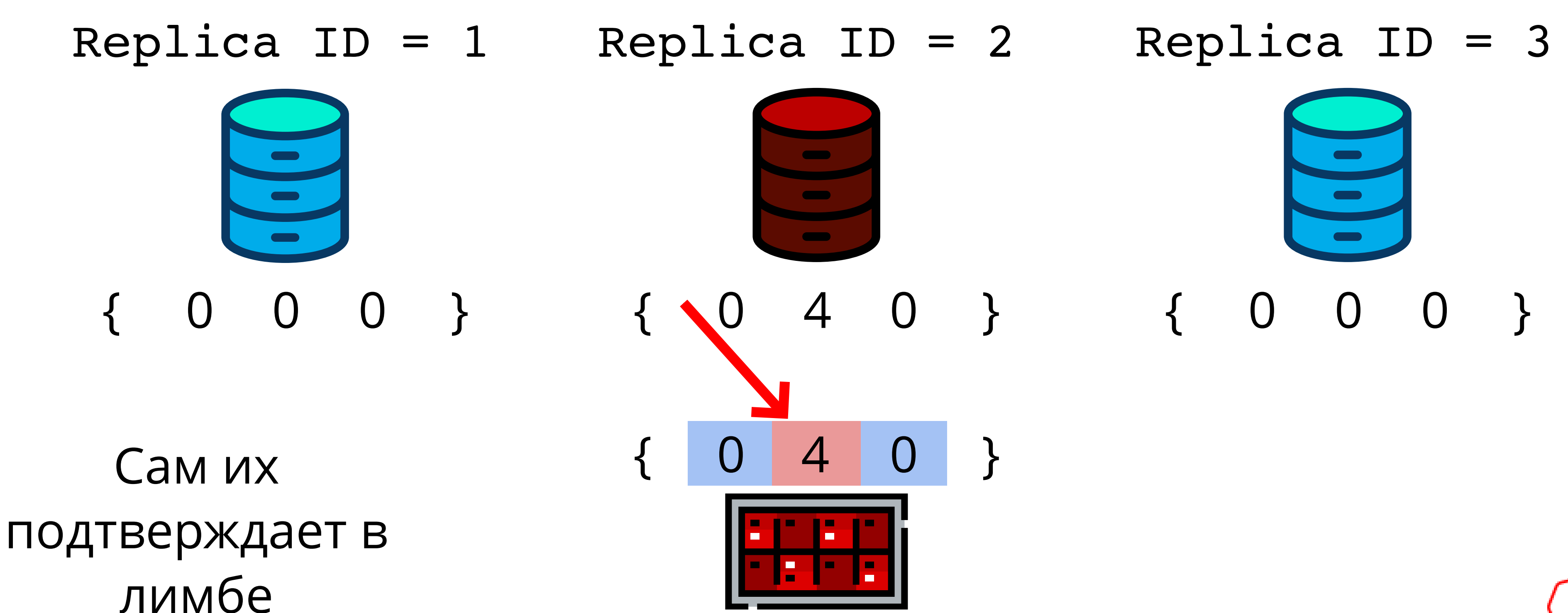
- **Replica ID** – ID реплики
- **LSN** – последний LSN **лидера,** подтвержденный этой **репликой**



Ожидание кворума [3]: сборка

Лимб использует особый vclock

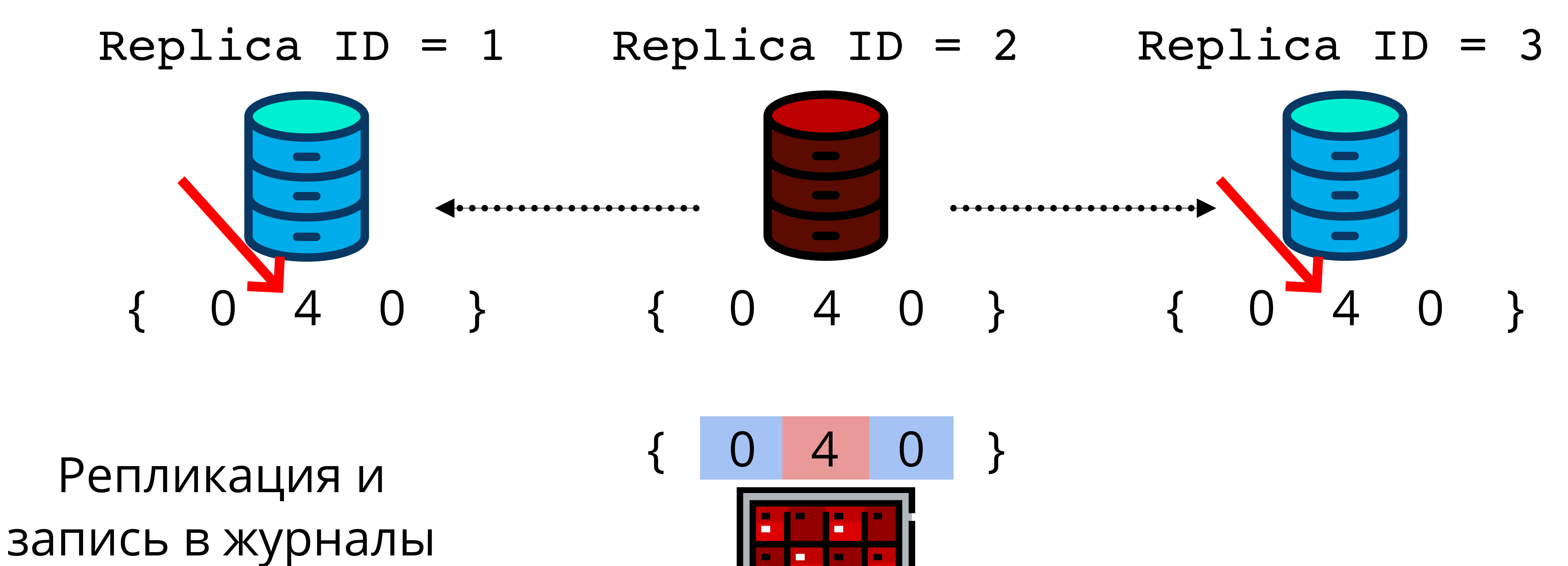
- **Replica ID** – ID реплики
- **LSN** – последний LSN **лидера,** подтвержденный этой **репликой**



Ожидание кворума [3]: сборка

Лимб использует особый vclock

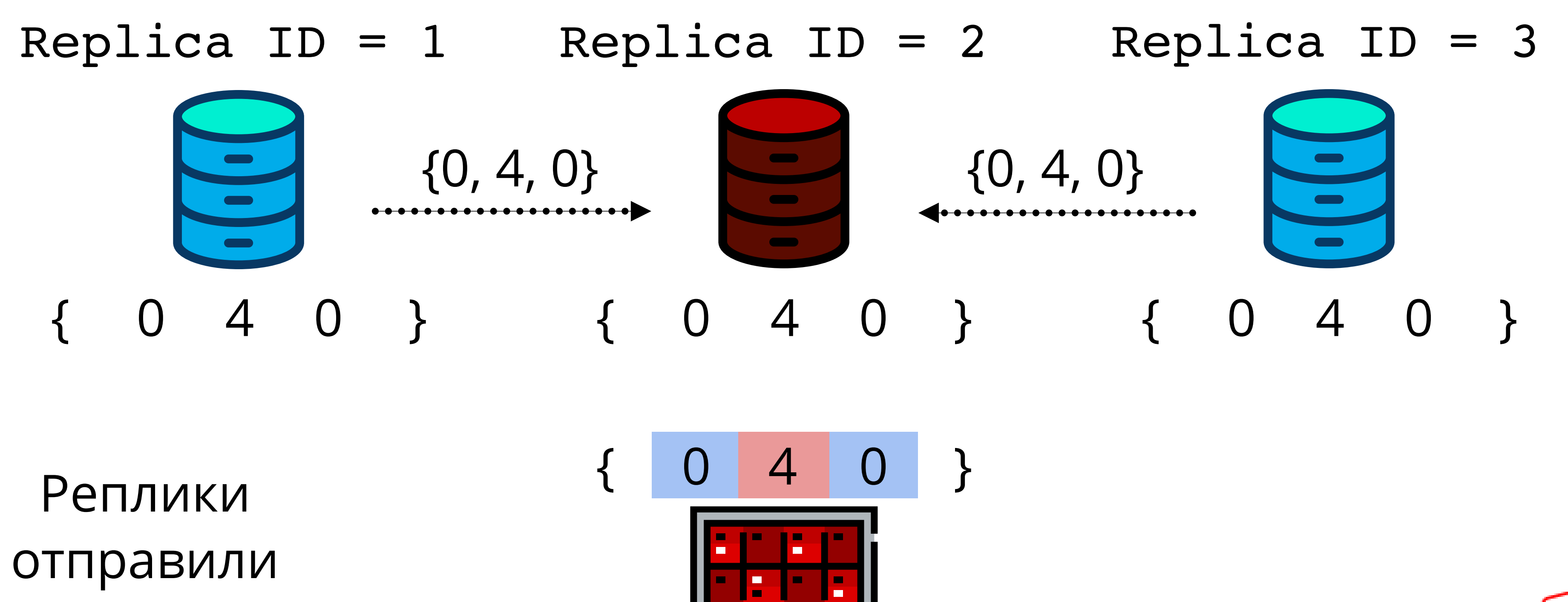
- **Replica ID** – ID реплики
- **LSN** – последний LSN **лидера,** подтвержденный этой **репликой**



Ожидание кворума [3]: сборка

Лимб использует особый vclock

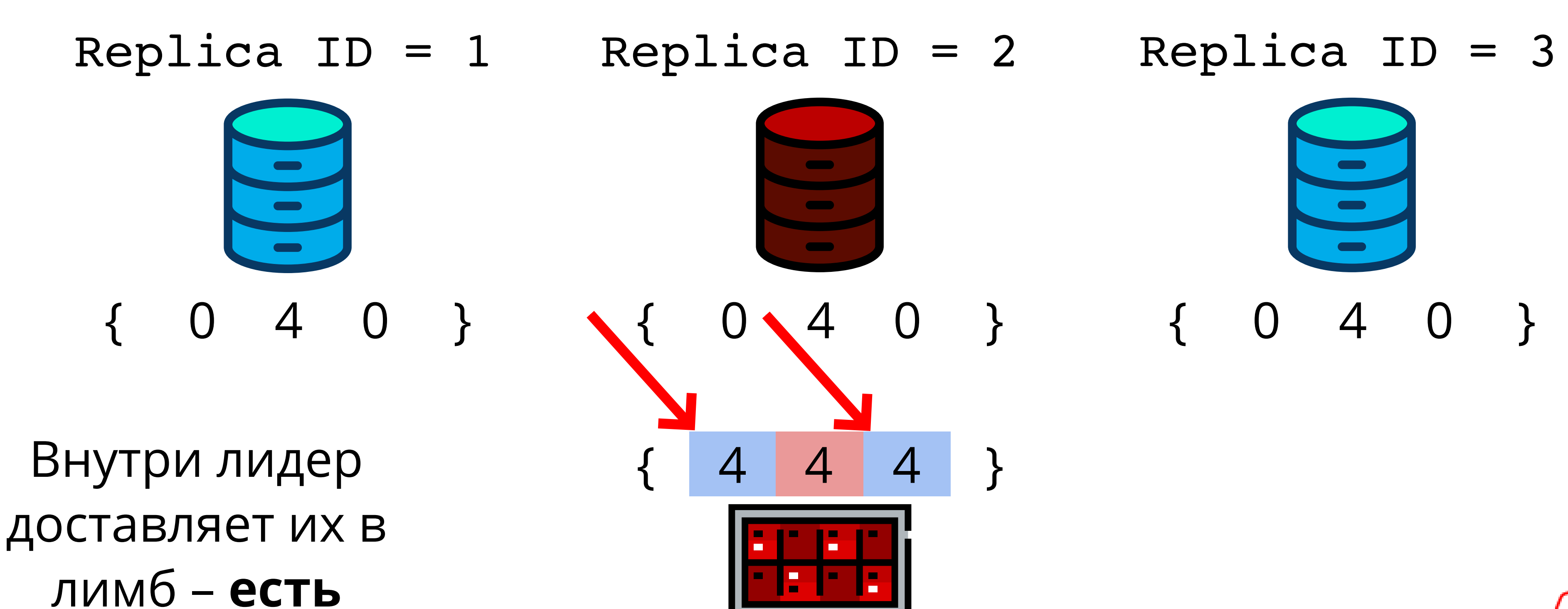
- **Replica ID** – ID реплики
- **LSN** – последний LSN **лидера,** подтвержденный этой **репликой**



Ожидание кворума [3]: сборка

Лимб использует особый vclock

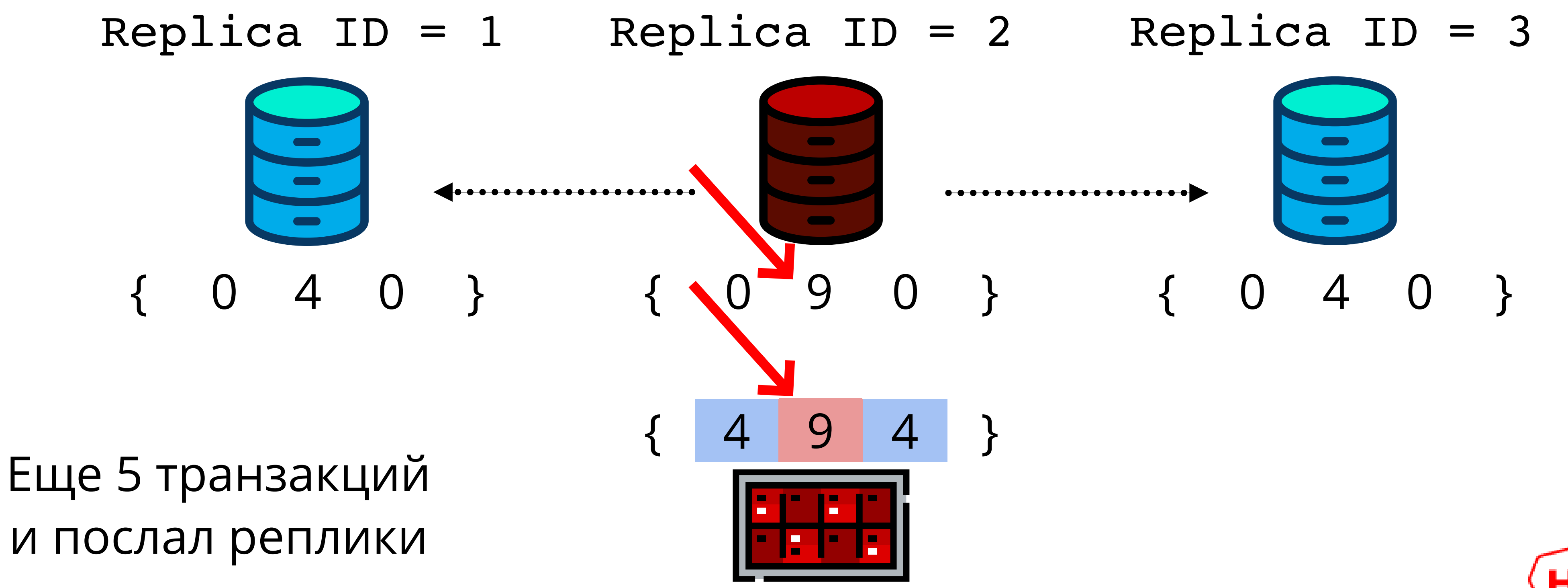
- **Replica ID** – ID реплики
- **LSN** – последний LSN **лидера,** подтвержденный этой **репликой**



Ожидание кворума [3]: сборка

Лимб использует особый vclock

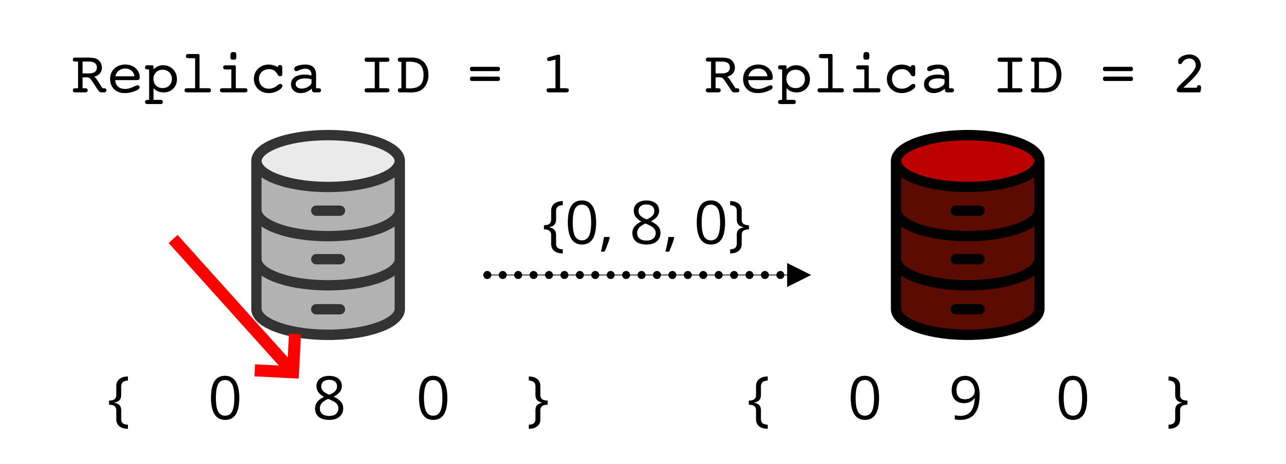
- **Replica ID** – ID реплики
- **LSN** – последний LSN **лидера,** подтвержденный этой **репликой**

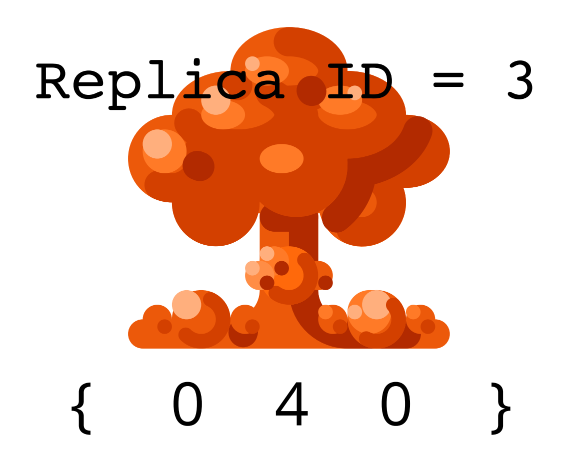


Ожидание кворума [3]: сборка

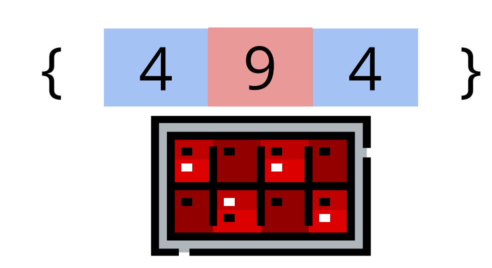
Лимб использует особый vclock

- **Replica ID** – ID реплики
- **LSN** – последний LSN **лидера,** подтвержденный этой **репликой**





{ 4 9 4 }

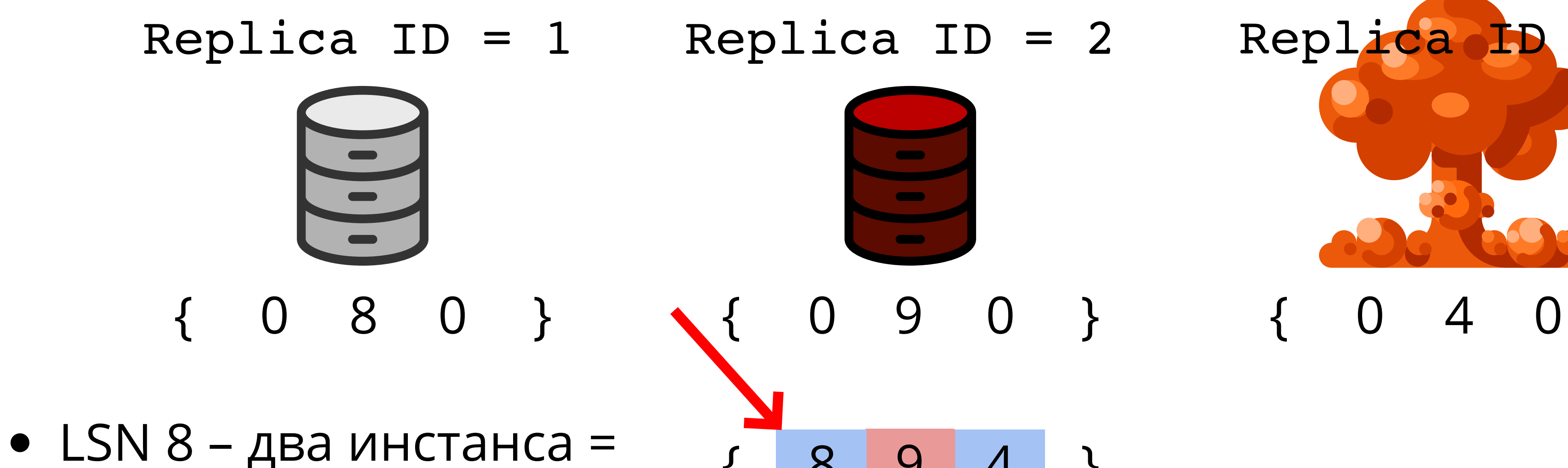


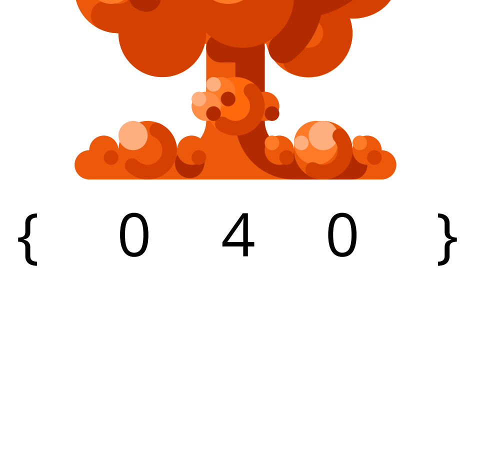
Одна реплика отказала, другая ответила частично

Ожидание кворума [3]: сборка

Лимб использует особый vclock

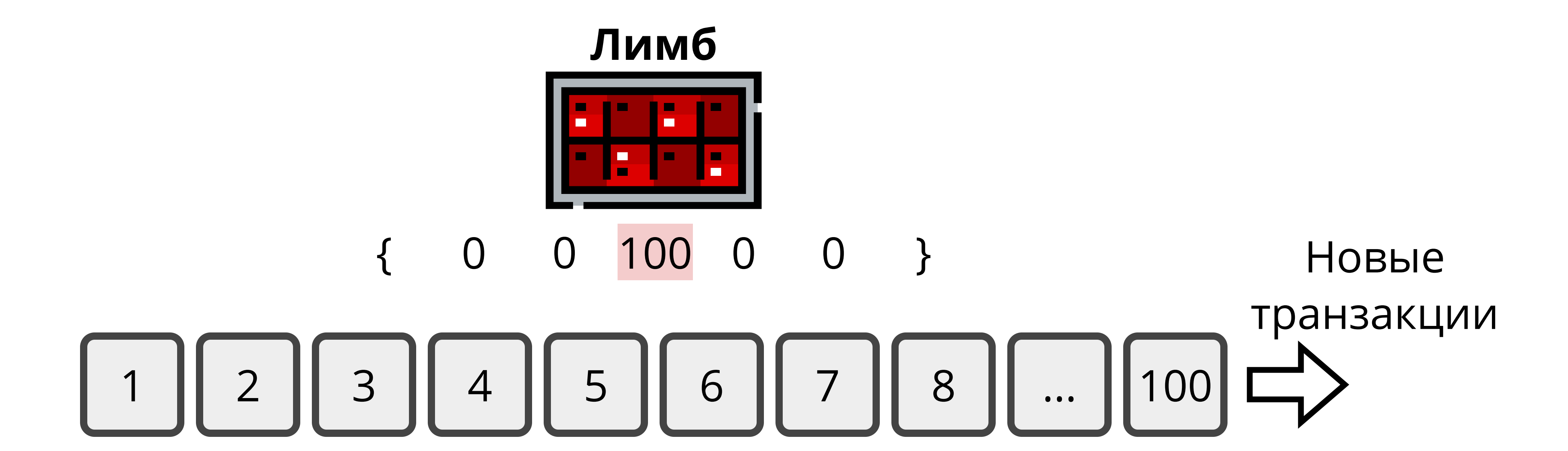
- **Replica ID** – ID реплики
- **LSN** – последний LSN **лидера,** подтвержденный этой **репликой**



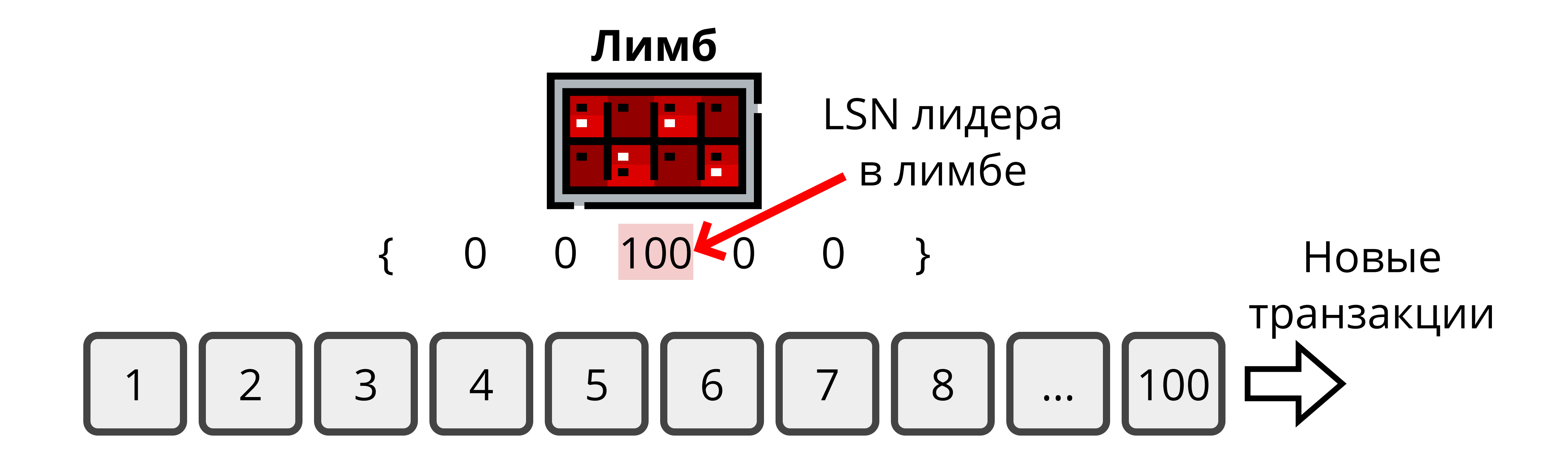


- LSN 8 – два инстанса = **кворум**
- LSN 9 – один инстанс = **нет кворума**

Коммит синхронной транзакции

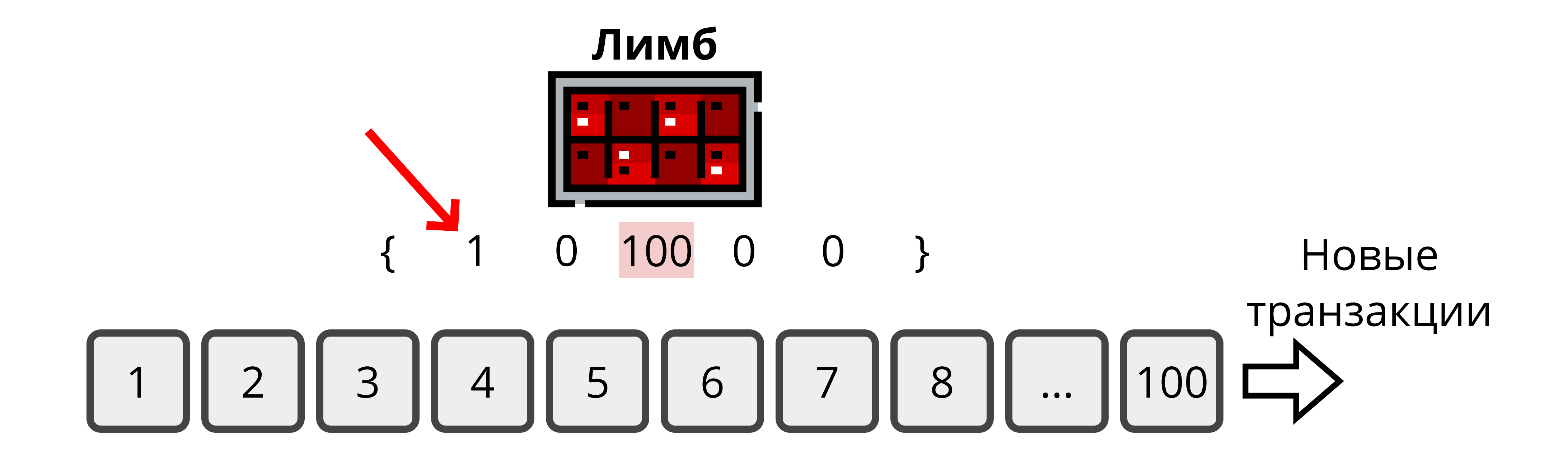


Коммит синхронной транзакции



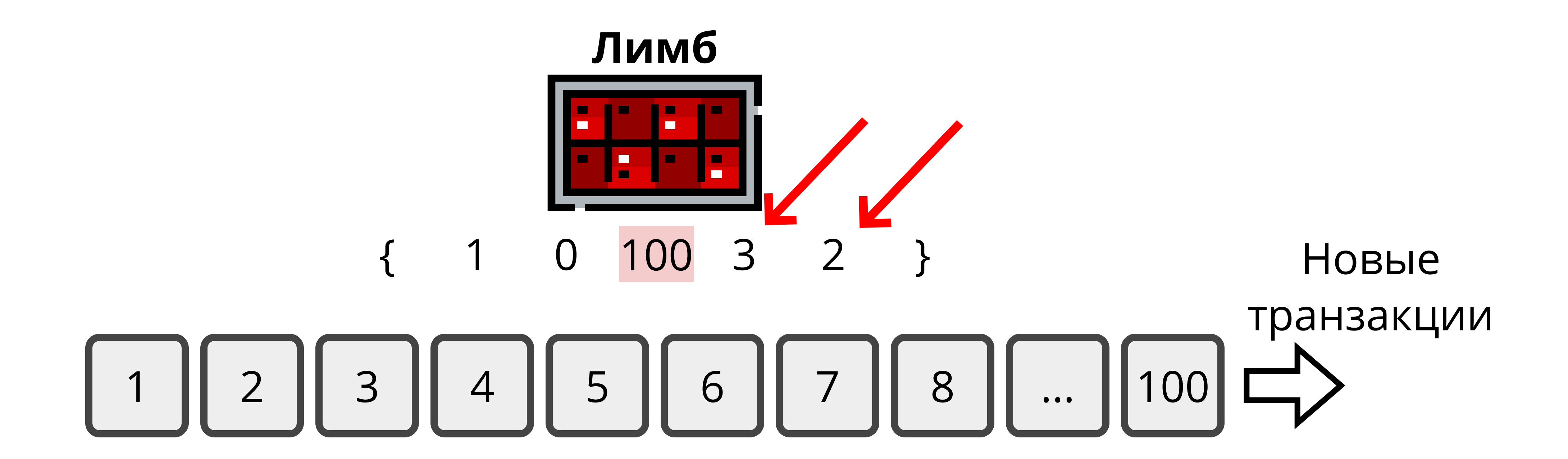
Все транзакции уже в журнале лидера

Коммит синхронной транзакции



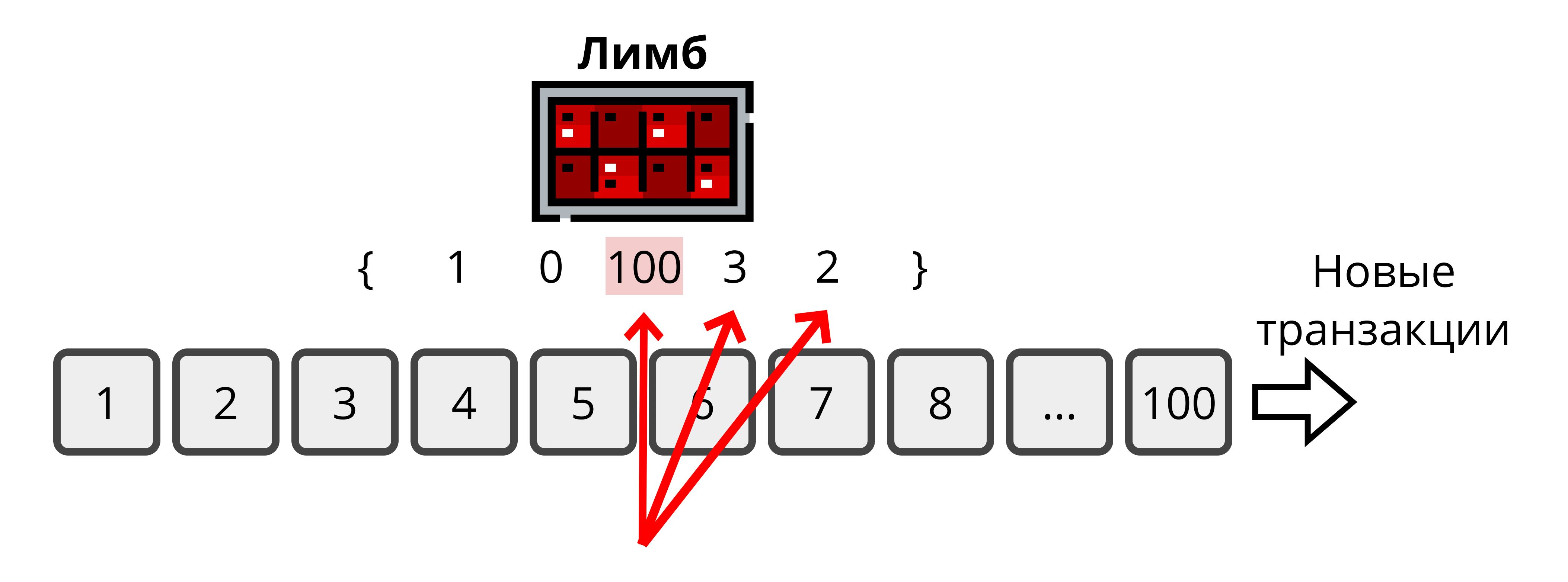
Пришло одно подтверждение – кворума нет

Коммит синхронной транзакции



Еще два подтверждения от других реплик

Коммит синхронной транзакции



Есть кворум на **LSN 2**:
от трех инстансов

Коммит синхронной транзакции

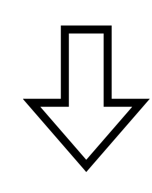
Лимб

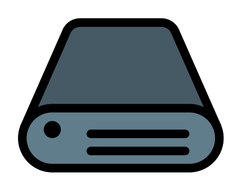
{ 1 0 100 3 2 }

Новые транзакции

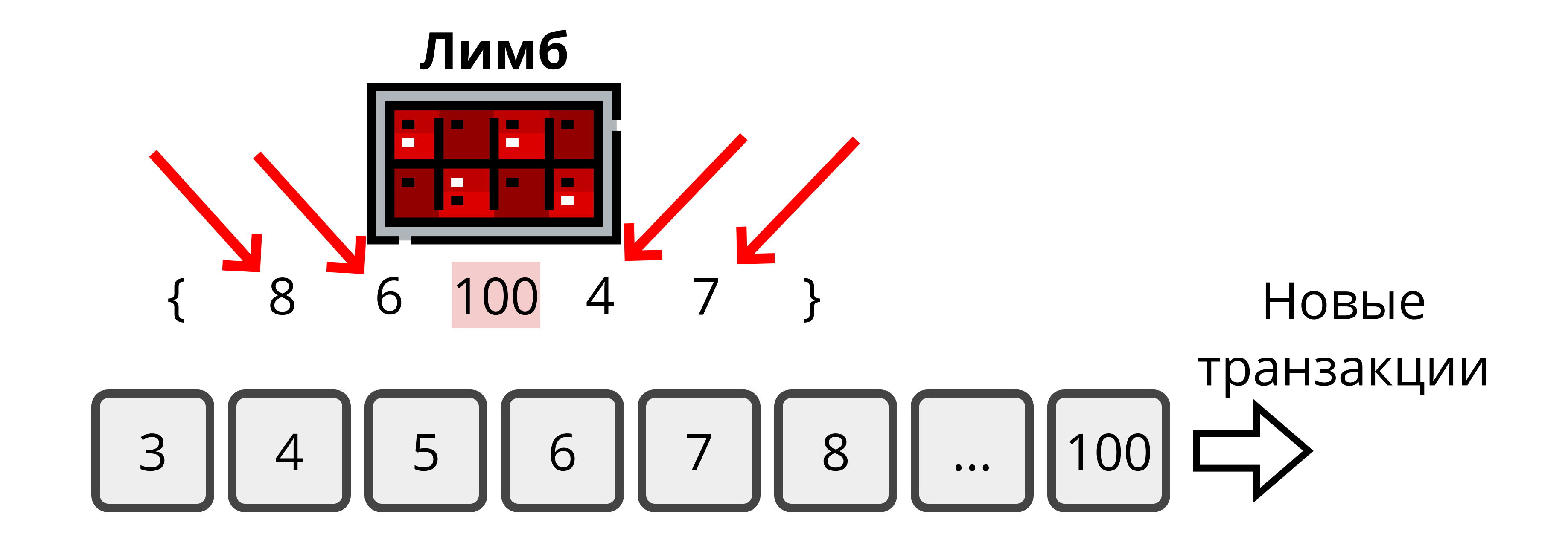
3	4	5	6	7	8	...	100

CONFIRM LSN 2



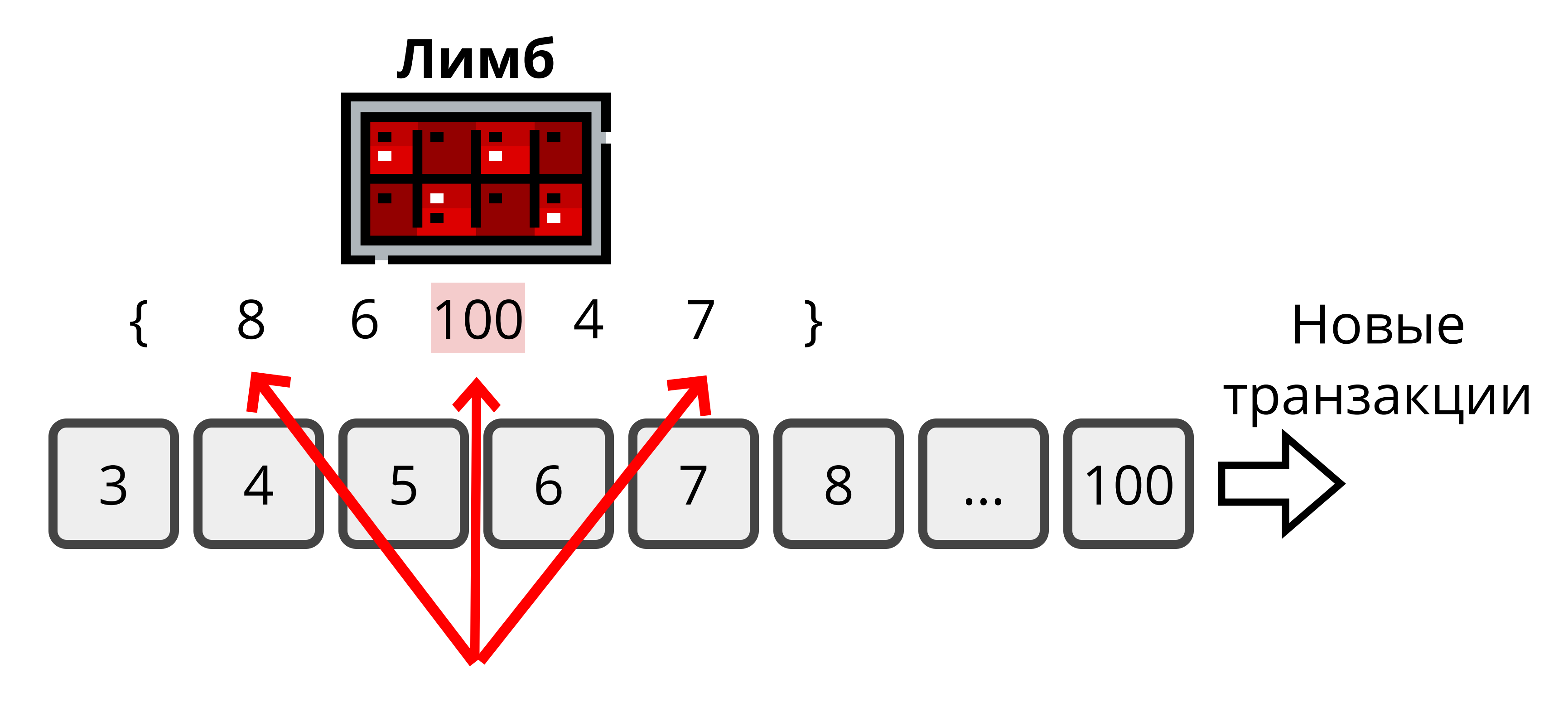


Коммит синхронной транзакции

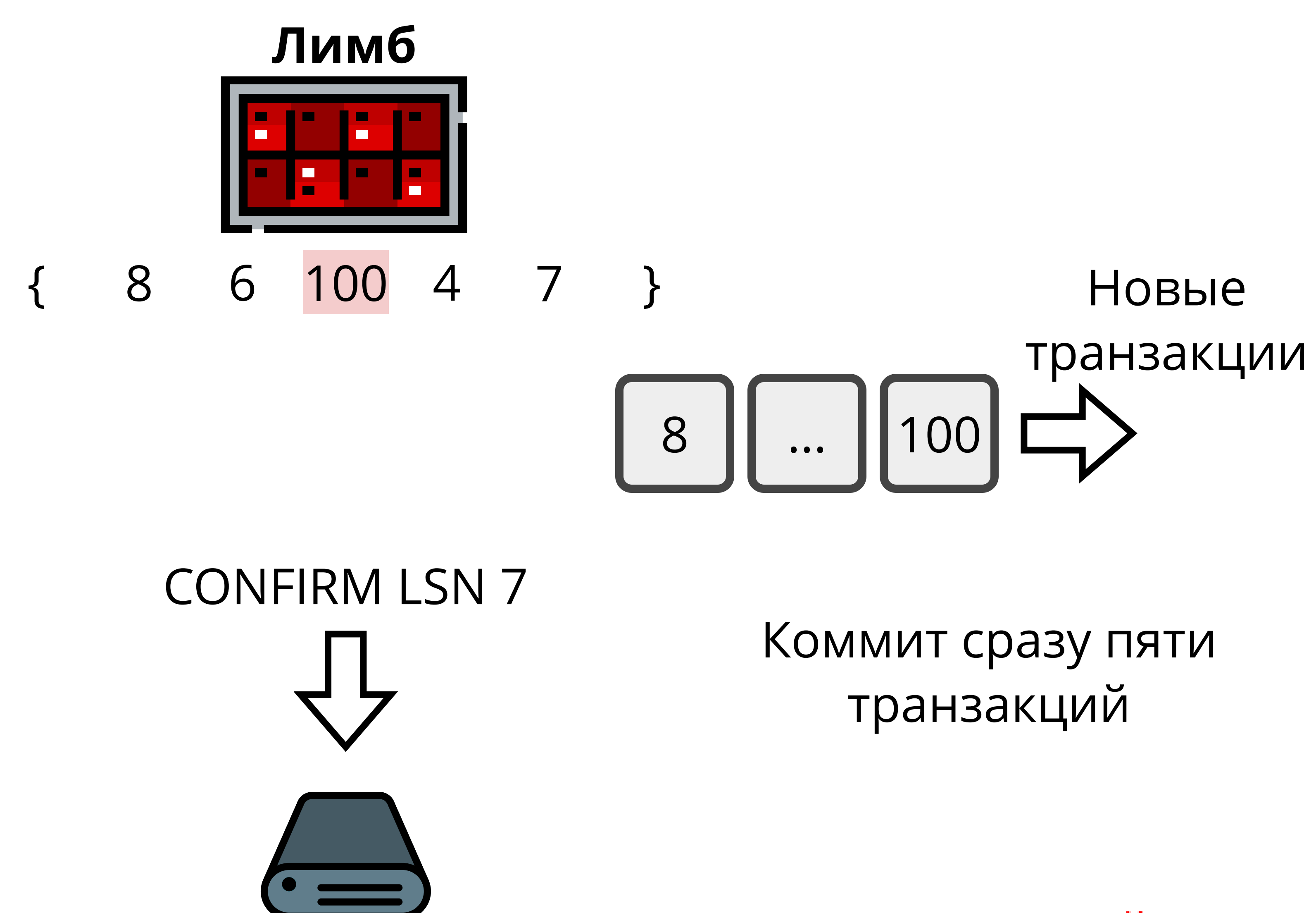


Еще несколько подтверждений

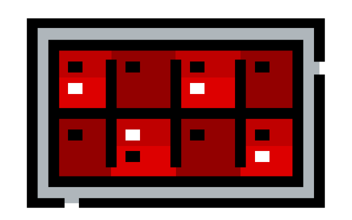
Коммит синхронной транзакции



Коммит синхронной транзакции



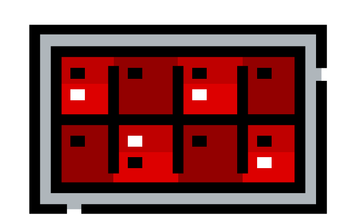
Откат синхронной транзакции [1]



Против бесконечного роста очереди есть **тайм-аут**

1	2	3	4	5	6	7

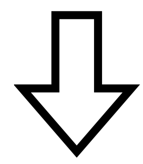
Откат синхронной транзакции [1]

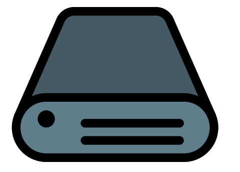


Против бесконечного роста очереди есть **тайм-аут**

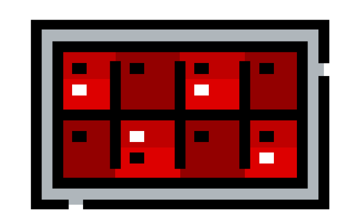
По истечении удаляются **все** транзакции, в журнал пишется ROLLBACK, пользователи получают ошибку

ROLLBACK LSN 1





Откат синхронной транзакции [1]

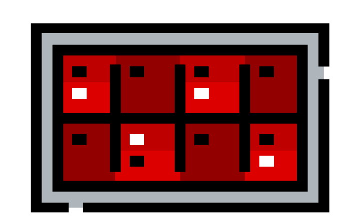


Против бесконечного роста очереди есть **тайм-аут**

По истечении удаляются **все** транзакции, в журнал пишется ROLLBACK, пользователи получают ошибку

Удаляются все, так как могут быть **зависимы по данным**

Откат синхронной транзакции [1]



Против бесконечного роста очереди есть **тайм-аут**

По истечении удаляются **все** транзакции, в журнал пишется ROLLBACK, пользователи получают ошибку

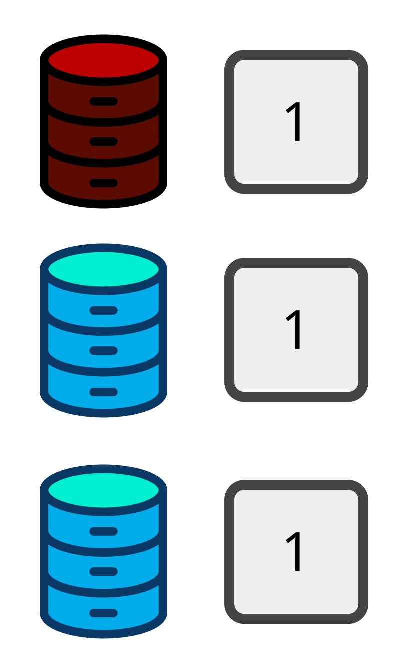
Удаляются все, так как могут быть **зависимы по данным**

Но **ROLLBACK** не гарантирует ничего! В случае перевыборов может произойти коммит

Откат синхронной транзакции [2]

ROLLBACK не дает гарантий!

Пусть есть транзакция на лидере

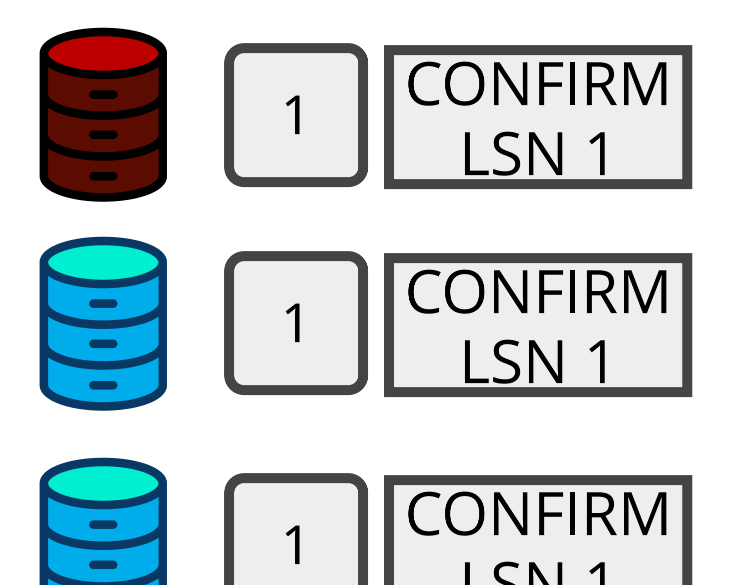


HighLoad++ Весна 2021

Откат синхронной транзакции [2]

ROLLBACK не дает гарантий!

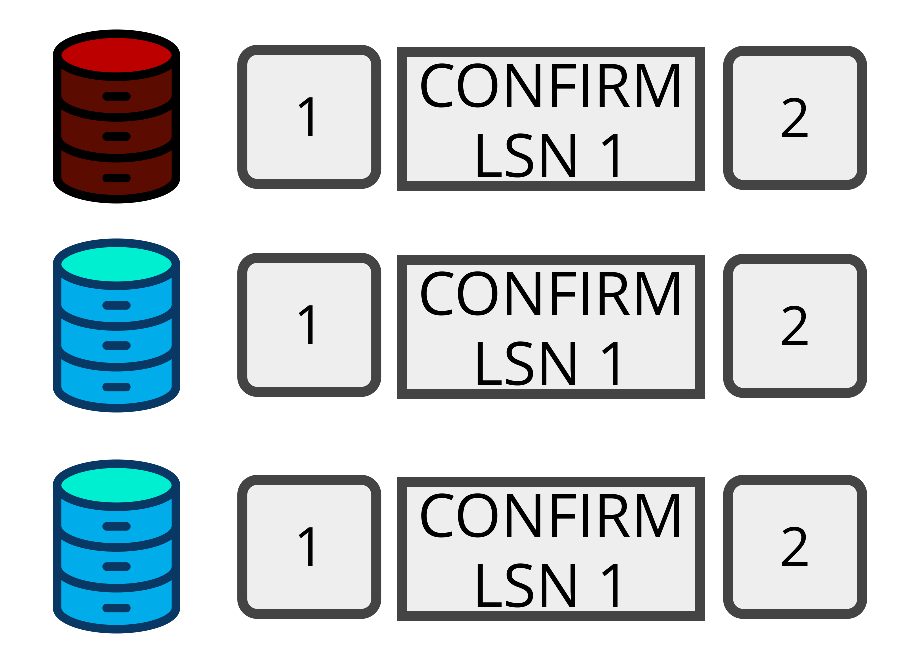
Кворум собран, коммит записан и разослан



Откат синхронной транзакции [2]

ROLLBACK не дает гарантий!

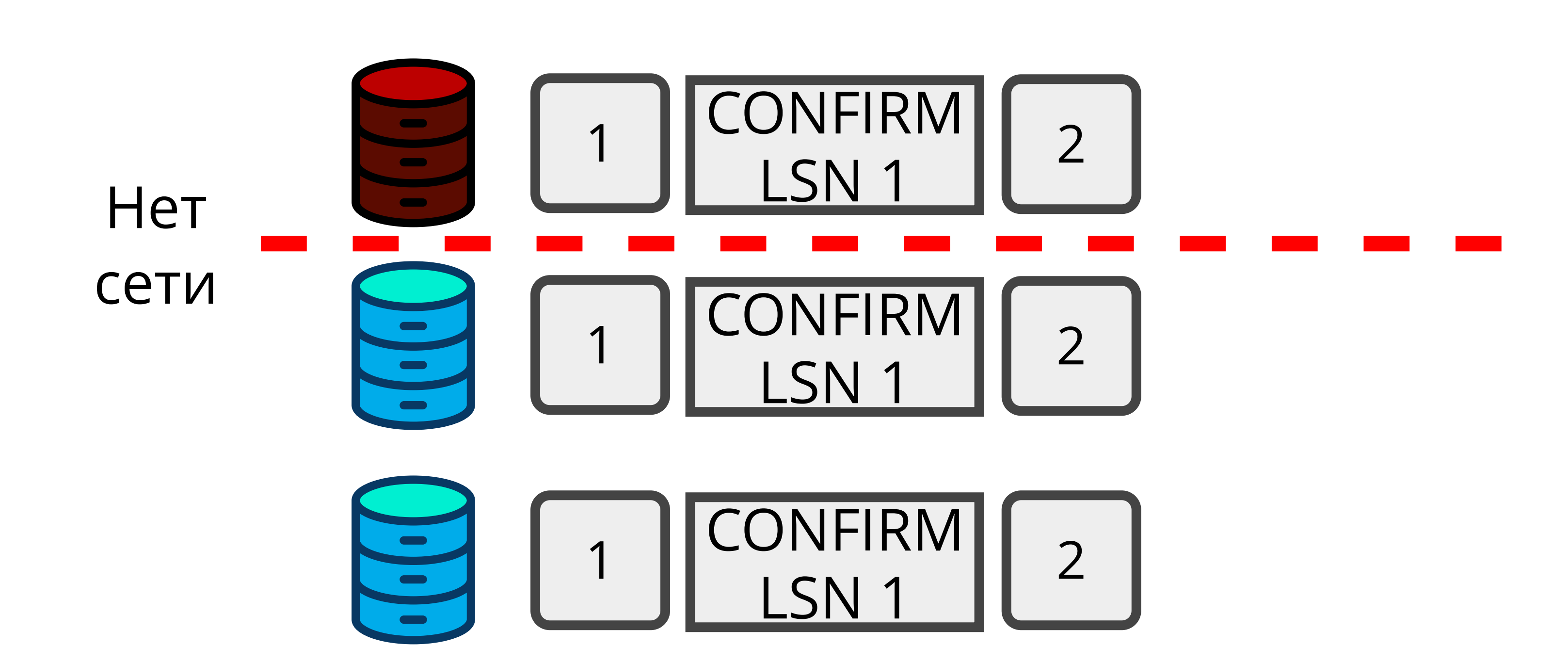
Еще транзакция и репликация



Откат синхронной транзакции [2]

ROLLBACK не дает гарантий!

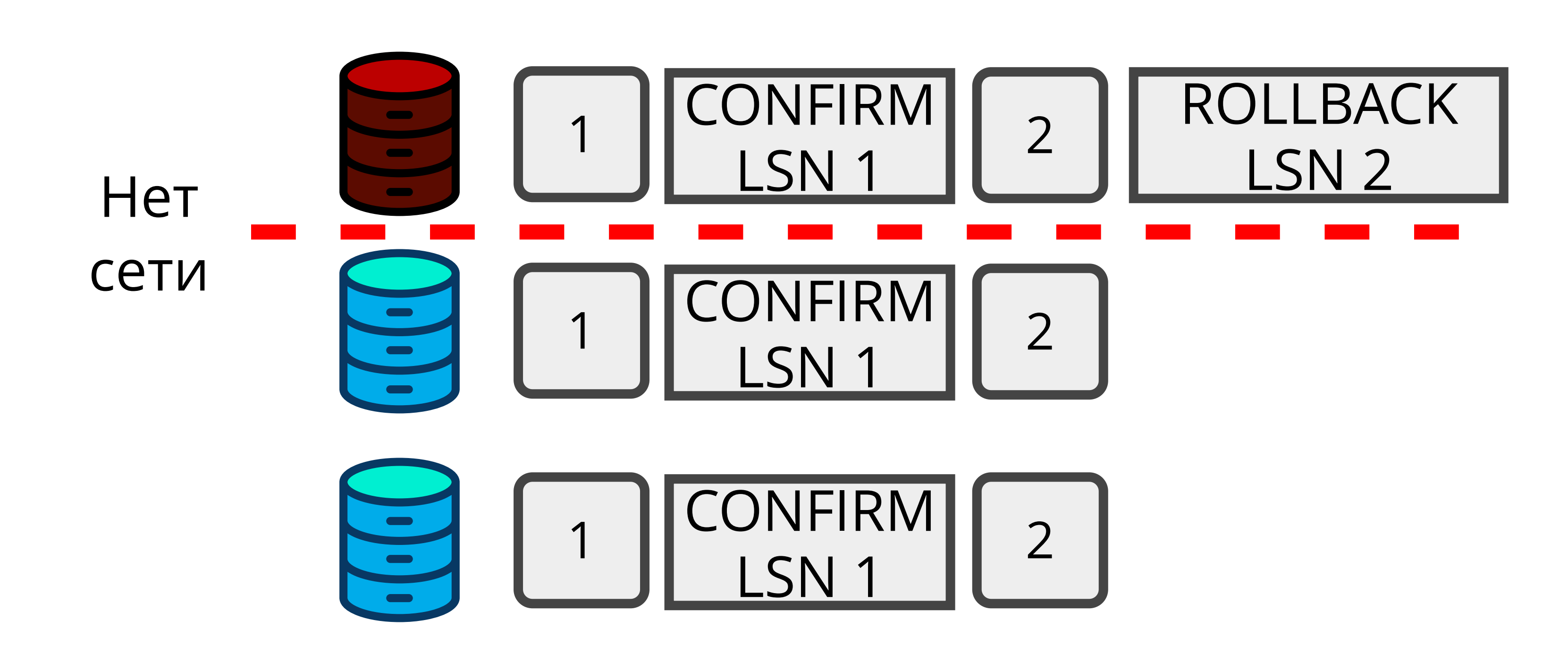
Сеть **сломалась**! Подтверждения не дошли



Откат синхронной транзакции [2]

ROLLBACK не дает гарантий!

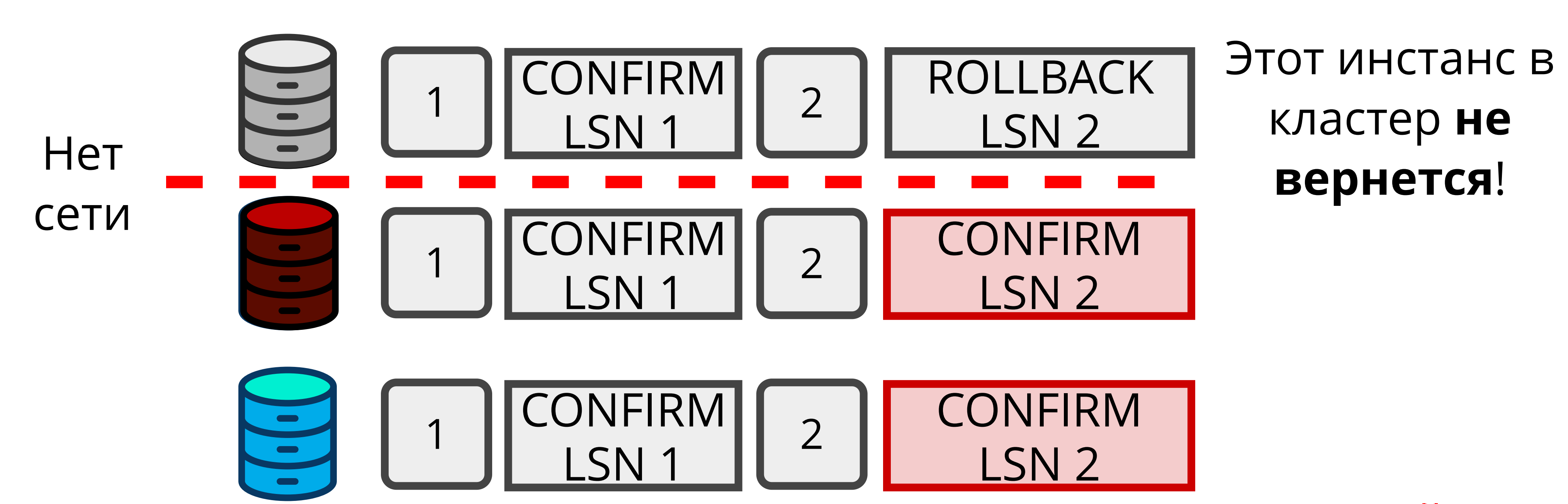
Мастер делает откат, **пользователь получил ошибку**



Откат синхронной транзакции [2]

ROLLBACK не дает гарантий!

Выбран **новый** лидер, и **сделал коммит**



API

```
box.cfg{
    replication_synchro_timeout = <seconds>,
    replication_synchro_quorum = <count>
}
```

HL HighLoad++ Весна 2021

API

Таймаут на сбор подтверждений

```
box.cfg{
    replication_synchro_timeout = <seconds>,
    replication_synchro_quorum = <count>
}
```

HighLoad++ Весна 2021

API

Таймаут на сбор подтверждений

```
box.cfg{
    replication_synchro_timeout = <seconds>,
    replication_synchro_quorum = <count>
}
```

<u>Кворум на коммит</u>

```
replication_synchro_quorum = "N/2 + 1"
replication_synchro_quorum = "math.max(2*N/3, 1)"
replication_synchro_quorum = 5
```

API

Таймаут на сбор подтверждений

```
box.cfg{
    replication_synchro_timeout = <seconds>,
    replication_synchro_quorum = <count>
}
```

Кворум на коммит

```
replication_synchro_quorum = "N/2 + 1"
replication_synchro_quorum = "math.max(2*N/3, 1)"
replication_synchro_quorum = 5
```

Создать синхронный спейс

```
box.schema.create_space(name, {
    is_sync = true
})
```

API

Таймаут на сбор подтверждений

```
box.cfg{
    replication_synchro_timeout = <seconds>,
    replication_synchro_quorum = <count>
}
```

Кворум на коммит

```
replication_synchro_quorum = "N/2 + 1"
replication_synchro_quorum = "math.max(2*N/3, 1)"
replication_synchro_quorum = 5
```

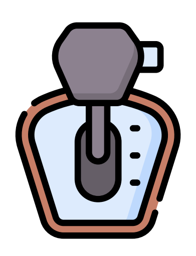
Создать синхронный спейс

```
box.schema.create_space(name, {
    is_sync = true
})
```

Мониторинг

```
box.info.synchro
---
- queue:
    len: 0
  quorum: 1
...
```

Автоматическая смена мастера

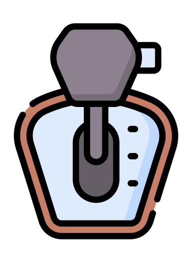


Работает по **Raft**

```
box.cfg{
    election_mode = <mode>,
    election_timeout = <seconds>,
    replication_synchro_quorum = <formula>,
}
```

HighLoad++ Весна 2021

Автоматическая смена мастера

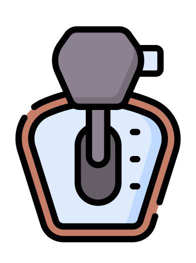


Работает по **Raft**

candidate / voter / off

```
box.cfg{
    election_mode = <mode>,
    election_timeout = <seconds>,
    replication_synchro_quorum = <formula>,
}
```

Автоматическая смена мастера



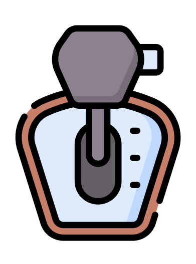
Работает по **Raft**

candidate / voter / off

```
box.cfg{
    election_mode = <mode>,
    election_timeout = <seconds>,
    replication_synchro_quorum = <formula>,
}
```

Таймаут выборов в секундах

Автоматическая смена мастера



Работает по **Raft**

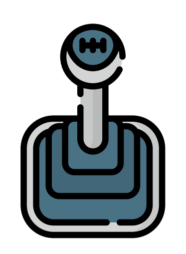
candidate / voter / off

```
box.cfg{
    election_mode = <mode>,
    election_timeout = <seconds>,
    replication_synchro_quorum = <formula>,
}
```

Таймаут выборов в секундах

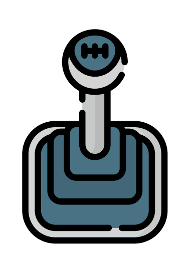
Кворум на выборы и репликацию

Ручная смена мастера



```
box.cfg{
    election_mode = <mode>,
    replication_synchro_quorum = <formula>,
}
```

Ручная смена мастера



manual / voter / off

```
box.cfg{
    election_mode = <mode>,
    replication_synchro_quorum = <formula>,
}
```

Кворум на выборы и репликацию

Назначение лидера

```
box.ctl.promote()
```

Пример [1]

Конфигурация мастера

```
box.cfg{
    listen = 3313,
    replication =  {
        '127.0.0.1:3313',
        '127.0.0.1:3314',
        '127.0.0.1:3315'
    },
    memtx_use_mvcc_engine = true,
    replication_synchro_quorum = 3,
    replication_synchro_timeout = 1000,
}
box.schema.user.grant('guest', 'super')
```

Пример [1]

Конфигурация мастера

```
box.cfg{
    listen = 3313,
    replication =  {
        '127.0.0.1:3313',
        '127.0.0.1:3314',
        '127.0.0.1:3315'
    },
    memtx_use_mvcc_engine = true,
    replication_synchro_quorum = 3,
    replication_synchro_timeout = 1000,
}
box.schema.user.grant('guest', 'super')
```

Две реплики

Пример [1]

Конфигурация мастера

```
box.cfg{
    listen = 3313,
    replication =  {
        '127.0.0.1:3313',
        '127.0.0.1:3314',
        '127.0.0.1:3315'
    },
    memtx_use_mvcc_engine = true,
    replication_synchro_quorum = 3,
    replication_synchro_timeout = 1000,
}
box.schema.user.grant('guest', 'super')
```

Две реплики

Выключить грязные чтения

Пример [1]

Конфигурация мастера

```
box.cfg{
    listen = 3313,
    replication =  {
        '127.0.0.1:3313',
        '127.0.0.1:3314',
        '127.0.0.1:3315'
    },
    memtx_use_mvcc_engine = true,
    replication_synchro_quorum = 3,
    replication_synchro_timeout = 1000,
}
box.schema.user.grant('guest', 'super')
```

Две реплики

Выключить грязные чтения

Кворум – 100%, для наглядности

Пример [1]

Конфигурация мастера

```
box.cfg{
    listen = 3313,
    replication =  {
        '127.0.0.1:3313',
        '127.0.0.1:3314',
        '127.0.0.1:3315'
    },
    memtx_use_mvcc_engine = true,
    replication_synchro_quorum = 3,
    replication_synchro_timeout = 1000,
}
box.schema.user.grant('guest', 'super')
```

Две реплики

Выключить грязные чтения

Кворум – 100%, для наглядности

Не париться про доступы

Пример [2]

Конфигурация реплик

Реплика 1

```
box.cfg{
    listen = 3314,
    replication = {
        '127.0.0.1:3313',
        '127.0.0.1:3314',
        '127.0.0.1:3315'
    },
    read_only = true,
    memtx_use_mvcc_engine = true
}
```

Реплика 2

```
box.cfg{
    listen = 3315,
    replication = {
        '127.0.0.1:3313',
        '127.0.0.1:3314',
        '127.0.0.1:3315'
    },
    read_only = true,
    memtx_use_mvcc_engine = true
}
```

Без опций *replication_synchro* – нужны только на мастере

Пример 3

Мастер	Реплика 1	Реплика 2

Пример 3

Мастер | Реплика 1 | Реплика 2

```
$> s = box.schema.create_space(
    'test', {is_sync = true})
$> _ = s:create_index('pk')
```

Создание схемы

Пример 3

Мастер

```
$> s = box.schema.create_space(
    'test', {is_sync = true})
$> _ = s:create_index('pk')

$> s:replace{1}
```

Проверка репликации

Реплика 1

Реплика 2

Пример 3

Мастер

```
$> s = box.schema.create_space(
    'test', {is_sync = true})
$> _ = s:create_index('pk')
$> s:replace{1}
```

Реплика 1

```
$> box.space.test:get({1})
---
- [1]
...
```

Реплика 2

```
$> box.space.test:get({1})
---
- [1]
...
```

Пример 3

Мастер

```
$> s = box.schema.create_space(
    'test', {is_sync = true})
$> _ = s:create_index('pk')
$> s:replace{1}
```

Реплика 1

```
$> box.space.test:get({1})
---
- [1]
...
```

Реплика 2

```
$> box.space.test:get({1})
---
- [1]
...
$> os.exit(1)
```

Реплика убита

☠

Пример 3

Мастер

```
$> s = box.schema.create_space(
    'test', {is_sync = true})
$> _ = s:create_index('pk')

$> s:replace{1}

$> fiber = require('fiber')
$> f = fiber.create(function()
    s:replace{2}
end)
$> f:status()
---
- suspended
...
```

Запуск синхронной транзакции в корутине

Реплика 1

```
$> box.space.test:get({1})
---
- [1]
...
```

Реплика 2

```
$> box.space.test:get({1})
---
- [1]
...
$> os.exit(1)
```

Реплика убита

☠

Пример [4]

Мастер

Реплика 1

Реплика 2

Пример [4]

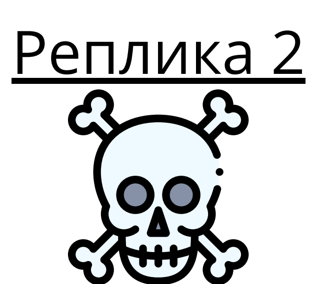
Мастер

```
$> s:get{2}
---
...
```

Реплика 1

```
$> box.space.test:get{2}
---
...
```

Реплика 2



Нет коммита – изменения не видны. На реплике тоже. Потому что **нет кворума**

Пример [4]

Мастер

```
$> s:get{2}
---
...
```

Реплика 1

```
$> box.space.test:get{2}
---
...
```

Реплика 2

```
$> box.cfg{...}
```

Перезапуск с тем же конфигом

Пример [4]

Мастер

```
$> s:get{2}
---
...

$> f:status()
---
- dead
...
```

Транзакция завершена на мастере

Реплика 1

```
$> box.space.test:get{2}
---
...
```

Реплика 2

```
$> box.cfg{...}
```

Пример [4]

Мастер

```
$> s:get{2}
---
...

$> f:status()
---
- dead
...
$> s:get{2}
---
- [2]
...
```

Реплика 1

```
$> box.space.test:get{2}
---
...

$> box.space.test:get{2}
---
- [2]
...
```

Реплика 2

```
$> box.cfg{...}

$> box.space.test:get{2}
---
- [2]
...
```

Изменения видны везде

Отличия от Raft

Векторный формат журнала

REDO-журнал

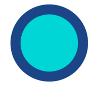
Чтение реплик

Откаты транзакций по времени



Отличия от Raft

Векторный формат журнала



Дорога к мастер-мастер-синхронности

REDO-журнал

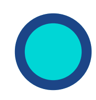
Чтение реплик

Откаты транзакций по времени



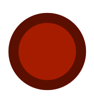
Отличия от Raft

Векторный формат журнала



Дорога к мастер-мастер-синхронности

REDO-журнал



Нельзя удалить лишние транзакции

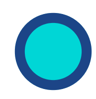
Чтение реплик

Откаты транзакций по времени



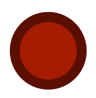
Отличия от Raft

Векторный формат журнала



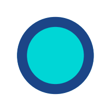
Дорога к мастер-мастер-синхронности

REDO-журнал



Нельзя удалить лишние транзакции

Чтение реплик



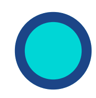
Выше доступность

Откаты транзакций по времени



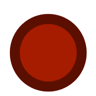
Отличия от Raft

Векторный формат журнала



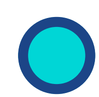
Дорога к мастер-мастер-синхронности

REDO-журнал



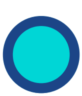
Нельзя удалить лишние транзакции

Чтение реплик



Выше доступность

Откаты транзакций по времени



Ограничение очереди



Планы

Повышение стабильности

И API может поменяться

Опции для транзакций

```
box.commit({is_lazy, is_sync})
```

Триггеры на выборы

```
box.ctl.on_election(function()
    ...
end)
```



The End









tarantool.io

